

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : Mohd Hafizuddin Bin Abd Shukor

Date of Birth : 18 March 1991

Title : Pulse Detonation Engine Controller and ADC Data Logging

Academic Session : **2013/2014**

I declare that this thesis is classified as:

☐**CONFIDENTIAL**

(Contains confidential information under the Official Secret Act 1972)*

☐**RESTRICTED**

(Contains restricted information as specified by the organization where research was done)*

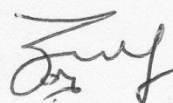
☒**OPEN ACCESS**

I agree that my thesis to be published as online open access (full text)

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:

1. The thesis is the property of Universiti Teknologi Malaysia
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

**SIGNATURE****SIGNATURE OF SUPERVISOR**

910318-12-5851

(NEW IC NO/PASSPORT)

Mr Zulfakar Bin Aspar

NAME OF SUPERVISOR

Date: 18 JUNE 2014

Date: 18 JUNE 2014

NOTES:

*

If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction.

"I hereby declare that I have read this thesis and in my/our*
opinion this thesis is sufficient in terms of scope and quality for the
award of the degree of Bachelor of Engineering (Electrical - Electronics)"

Signature

: 

Name of Supervisor

: Mr Zulfakar Bin Aspar

Date

: 18 JUNE 2014

PULSE DETONATION ENGINE CONTROLLER WITH ADC DATA LOGGING

MOHD HAFIZUDDIN BIN ABD SHUKOR

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Engineering (Electrical - Electronics)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2014

Specially dedicated to
all good Muslims on the earth
Honour people of Malaysia
And my beloved parents

I declare that this thesis entitled "*Pulse Detonation Engine Controller and ADC Data Logging*" is the result of my own research except as cited in the references.
The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature

:


.....

Name

:

HAFIZUDDIN BIN ABD SHUKOR

Date

:

18 June 2014

Specially dedicated to
all good Muslim on the earth,
Honest people of Malaysia
And my beloved parents
Dr Abd Shukor Bin Abd Hamid and Mrs Maisharah Md Said

ACKNOWLEDGEMENT

Thesis writing for Final Year Project (FYP) is a mark of quality and commitment in engineering studies as well as a persuasion for personal virtues and ethics such as honesty, self-confidence and continuous self-improvement. This task has led me to work with invaluable people of different backgrounds which have taught me a lot on how to become an excellent person and engineer. Particularly, I would like to thank my FYP supervisor, Mr Zulfakar Bin Aspar for his guidance and exposure on real-life industrialized engineering world. Moreover, millions of thanks for his constructive criticism which always tell my mind that becoming a very good engineer is not an easy job. Without his supervision, the experience in doing this project would be common and dull.

I also want to send my gratitude to the master students in Embedded System Laboratory which had helped me much in developing the project. Their experience and expertise related to my project is very much respected. Without them, the project will be delayed and going nowhere.

Last, I would also like to thank my fellow friends and colleagues that are also doing their project. Their tips and suggestion on improving my work somehow 'furnish' my project to become better.

ABSTRAK

Laporan ini menerangkan implementasi dan analisis pengawal serta mekanisme pengelogan data berdasarkan mikropengawal untuk operasi Enjin Denyut Letupan. Sistem dibangun menggunakan mikropengawal LPC1769 yang berdasarkan rekabentuk piawai industri ARM Cortex-M3. Tujuan implementasi sistem ini adalah untuk membolehkan Enjin Denyut Letupan dioperasikan pada kelajuan tinggi secara tepat semasa data operasi dilog ke dalam peranti simpanan memori luaran. Sistem mengimplementasikan algoritma operasi Enjin Denyut Letupan menggunakan periferal Input Output Kegunaan Am (GPIO) dan pemasa 'Interrupt' mikropengawal LPC1769 berdasarkan Enjin Denyut Letupan sedia ada yang dibangunkan makmal Aliran Reaksi Kelajuan Tinggi di Fakulti Kejuruteraan Mekanikal (FKM), Universiti Teknologi Malaysia (UTM). Mekanisme pengelogan data diimplementasikan menggunakan pustaka sumber bebas Sistem Fail FAT (*File Access Table*) daripada Chan dintegrasikan dengan Penukar Analog Digital 12-bit pada LPC1769 untuk mengukur dan mengelog operasi Enjin Denyut Letupan ke dalam memori imbas USB. Prototaip akhir untuk modul pengawal berupaya melaksanakan operasi kawalan apabila diuji menggunakan litar penguji yang terdiri daripada susunan LED (Light Emitting Diode) dimana sela masa diantara setiap peringkat ulangan Enjin Denyut Letupan boleh dikonfigurasi sehingga ketepatan mikrosaat sementara modul pengelogan data boleh mengelog sehingga 88 sampel per saat.

ABSTRACT

This report describes the implementation and analysis of a controller and data logging mechanism based on microcontroller for the operation of Pulse Detonation Engine (PDE). The system was developed using LPC1769 microcontroller based on industrial standard ARM Cortex-M3 architecture. The purpose of implementing such system is to enable the engine to be operated at high speed accurately while the data on PDE operation is logged into external memory storage device. The system implements PDE operation algorithm using General Purpose Input Output (GPIO) and Interrupt timer peripheral of LPC1769 based on existing PDE prototype developed by High Speed Reaction Flow (HiREF) in Faculty of Mechanical Engineering, Universiti Teknologi Malaysia (UTM). The data logging mechanism was implemented using open source Chan's FAT (File Access table) File System library integrated with 12-bit ADC (Analog Digital Converter) on-board LPC1769 to measure and log PDE operation data into USB flash memory. The final prototype for controller module was able to perform control operation when tested with a test circuit consist of LED (Light Emitting Diode) array where the timing interval between each stage of PDE operation cycle can be configured until microsecond accuracy while the data logging module can log up to 88 samples per second.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	TITLE	i
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	xi
	LIST OF FIGURES	xii
	LIST OF ABBREVIATIONS	xiv
1	INTRODUCTION	1
	1.1 Project Background	1
	1.2 Problem Statement	2
	1.3 Project Objective	3
	1.4 Scope of Work	4
2	LITERATURE REVIEW	5
	2.1 Pulse Detonation Engine (PDE)	5
	2.1.1 PDE Theory and Concepts	5
	2.1.2 PDE Efficiency	6
	2.1.3 Issues On Practical Implementation Of PDE	7
	2.2 Rocket Engine Control System	7
	2.3 FKM's HiREF Laboratory Pulse Detonation Engine	9
	2.3.1 Pulse Detonation Engine Structure	9
	2.3.2 Pulse Detonation Engine Operation	11

2.3.3	Pulse Detonation Engine Control Circuit	11
2.3.4	Measurement Sensors	12
2.4	LPC1769 Microcontroller Unit (MCU)	13
2.4.1	LPCXpresso LPC1769 Development Board	14
2.4.2	LPCXpresso IDE (Integrated Development Environment)	14
2.5	Serial Communication	15
2.5.1	UART Interface on Microcontroller	15
2.5.2	Serial Communication Interface Using Computer	16
3	PROJECT METHODOLOGY	17
3.1	Project Methodology Flow Chart	17
3.2	Top Level Design	19
3.3	Serial Communication Interface	20
3.3.1	UART-USB Converter	20
3.4	Pulse Detonation Engine Controller Design	21
3.4.1	Isolator Circuit	22
3.4.2	PDE Controller Operation	24
3.4.3	PDE Controller Mechanism & Firmware Design	25
3.5	ADC Data Logging	27
3.5.1	ADC Signal Attenuator	27
3.5.2	External Memory Storage	28
3.5.3	ADC Data Logging Firmware	30
3.6	Testing and Verification	33
3.6.1	PDE Controller Testing and Verification	33
3.6.2	ADC Data Logging Testing and Verification	34
3.6.2.1	ADC Data Logging Firmware Timing Analysis Method	34

	3.6.2.2 ADC Data Logging	35
	Performance Analysis Method	
4	RESULTS & DISCUSSION	36
4.1	PDE Controller Analysis	36
4.1.1	LED Test Circuit Limitation	37
4.2	ADC Data Logging Analysis	37
4.2.1	Signal Attenuator Linearity Analysis	37
4.2.2	Data Logging Timing Analysis	39
4.2.2	Data Logging Performance Analysis	40
4.2.3	UART-based Data Capture Performance	44
	Analysis & Comparison	
4.3	Final Project Prototype	45
5	PROJECT MANAGEMENT	47
5.1	Market Survey	47
5.1.1	Need	47
5.1.2	Approach	48
5.1.3	Benefit Per Cost	48
5.1.4	Competition	48
5.2	Time Management	48
5.3	Sourcing Management	49
5.3.1	Testing Equipment	49
5.3.2	Electronic Components/Tools	50
5.3.3	Firmware Development	52
5.3.4	End-User Consultation	53
5.3.5	Teammate/Colleague	53
5.4	Financial Management	54
6	CONCLUSION & SUGGESTION	56
6.1	Conclusion	56
6.2	Suggestion for Future Improvement	56

REFERENCE

58

Appendix 1

59

LIST OF TABLES

TABLE NO.	TITLE	PAGE
3.1	Tabulation of actuators and control voltage	21
3.2	Resistor and supply voltage value for isolator circuit	23
3.3	Example timing intervals with respect to PDE operation frequency	25
3.4	Chan's FAT File System function in sequence	31
4.1	Attenuator configuration 1 result	37
4.2	Attenuator configuration 2 result	38
4.3	Calibration comparison between before and after attenuation	39
4.4	Data logging position markings	39
4.5	Timing analysis at random data line	39
4.6	Summary of data logging performance analysis	44
5.1	Equipment sourcing tabulation	50
5.2	Bill of material (BOM) for this project	52
5.3	Components/Tools readily available	52
5.4	Hardware development costing	55

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	An example of pulse detonation engine structure	2
1.2	Basic operation cycle for pulse detonation engine	3
2.1	SSME thrust control system (with close-loop feedback)	9
2.2	Actual PDE prototype developed by HiREF	10
2.3	CNG injector from LO-Gas	10
2.4	Time interval between each stage	11
2.5	Ignition circuit	12
2.6	(a) Pressure transducer, (b) accelerometer, (c) load cell	12
2.7	LPCXpresso LPC1769 development board	13
2.8	LPCXpresso IDE workspace	14
2.9	RS232 Loopback Testing (shorting red coloured pins)	15
3.1	Core Project Methodology	18
3.2	Top level graphical representation of the project	19
3.3	UC00B V2011, Cytron's USB-UART converter	21
3.4	Optocoupler internal schematic	22
3.5	Isolator circuit schematic (for all 8 control pins)	23
3.6	PDE controller flow chart	26
3.7	Attenuator (Configuration 1)	28
3.8	Attenuator (Configuration 2)	28
3.9	USB Host implementation for LPC1769	29
3.10	USB Host actual implementation (using breadboard)	29
3.11	ADC data logging operation flowchart	32
3.12	Test circuit schematic for PDE controller	33
3.13	Actual test circuit for PDE controller	34
3.14	Timing analysis framework	35
4.1	PDE controller operation using test circuit	36
4.2	Attenuator configuration 1 plot	38

4.3	Attenuator Configuration 2 Plot	38
4.4	Comparison of multiple sector and single sector per block writing	40
4.5	Data logging file example	41
4.6	Test 1 ADC data logging result	41
4.7	Test 2 ADC data logging result	42
4.8	Test 3 ADC data logging result	42
4.9	Test 4 ADC data logging result	43
4.10	Test 5 ADC data logging result	43
4.11	ADC data capture using <i>HyperTerminal</i>	45
4.12	Final project prototype	46
5.1	Working Timeline for FYP1	49
5.2	Working Timeline for FYP2	49

LIST OF ABBREVIATIONS

PDE	- Pulse Detonation Engine
NASA	- National Aeronautic and Space Administration
UTM	- Universiti Teknologi Malaysia
FKM	- Fakulti Kejuruteraan Mekanikal
HiREF	- High Speed Reacting Flow
IDE	- Integrated Development Environment
ADC	- Analog Digital Converter
MCU	- Microcontroller Unit
TTL	- Transistor-Transistor Logic
NABC	- Needs, Approach, Benefit per cost, Competition
USB	- Universal Serial Bus
CAN	- Controller Area Network
SPI	- Serial Peripheral Interface
UART	- Universal Asynchronous Receiver & Transmitter
SWD	- Serial Wire Debug
GUI	- Graphical User Interface
rpm	- Rotation per minute
ms	- mille second
CNG	- Compressed Natural Gas
SSME	- Space Shuttle Main Engine
PWM	- Pulse Width Modulation
GPIO	- General Purpose Input Output
PDIP	- Plastic Dual In-Line Package
SOIC_N	- Standard Small Outline Package
SMD	- Surface Mounted Device
IC	- Integrated Circuit

CHAPTER 1

INTRODUCTION

1.1 Project Background

Detonation Engine (sometimes termed: Pulse Detonation Engine, PDE) is a very efficient hypersonic pulse combustion engine targeted for aeronautic applications [1]. It is still in research stage in many institutions including United States' NASA (National Aeronautics and Space Administration) and Universiti Teknologi Malaysia (UTM). The engine is functioning by using the power administered from the detonation of fuel, instead of typical technology which is combustion of fuel that will transfer the energy to rotate the shafts and move the pistons. Until now, UTM has its own Pulse Detonation Engine being developed under Faculty of Mechanical Engineering (FKM) High Speed Reacting Flow (HiREF) laboratory.

By referring to Figure 1.1, the key concept of the engine is to create pulsating detonation waves in the detonation tube through detonation in fuel-oxidizer mixing chamber [2]. After the fuel and oxidizer is injected and mixed in a certain ratio inside mixing chamber, ignition will be initiated which results in a detonation. After the detonation ends, clean air will be purged inside the engine to remove burnt fuels. Blast from the detonation will produce enough thrust to drive particular aerial vehicles. The whole processes (fuel-oxidizer injection, ignition, purging) are considered as one cycle of engine operation and it will repeat for many cycles for continuous operation of the engine.

The control mechanism for the engine is basically divided into mechanical and electrical based actuators that control the fuel-oxidizer injection, ignition and purge phase of the engine operation [2]. Institutions throughout the globe are still competing on the most effective operation control and mechanism for this engine.

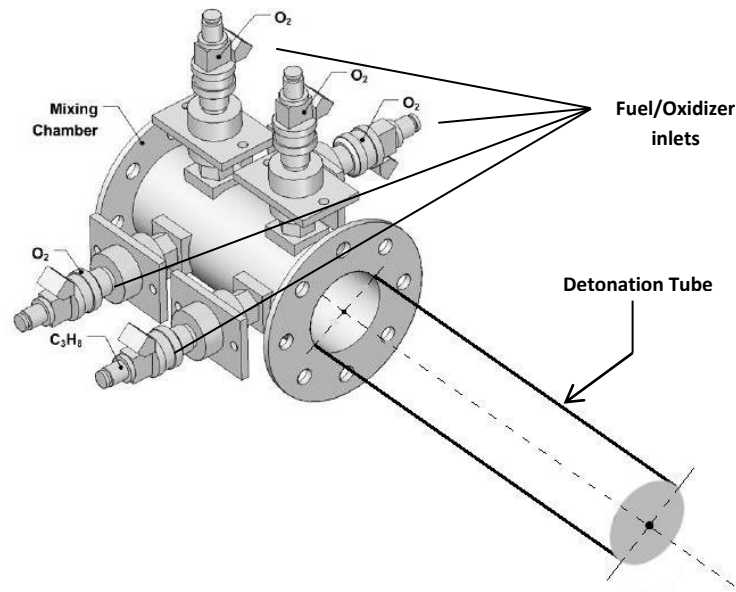


Figure 1.1 : An example of pulse detonation engine structure [1]

1.2 Problem Statement

Pulse Detonation Engine (PDE) concept mainly circles around its detonation process. The detonation process has a very significant property which is the speed of detonation cycles. Detonation-based engine cycles are so fast that the fuel and oxidizer injection phase time interval is much longer than the detonation period [1]. With the combination of injection (for fuel and oxidizer), ignition and purge phase time, the timing control of the controlling actuators becomes more complex and harder to realized. All of the phase needs to be accurately executed. Figure 1.2 shows the visualization for timing proportions of all operation phase in one cycle. This cycle should be repeated for continuous operation.

Moreover, the pulse nature of the detonation process need a very fast, accurate, stable and effective control mechanism in order to maintain the repeating

detonation cycles throughout the entire engine operation. The repeating detonation cycles were measured in frequency instead of rotation per minute (rpm) in conventional combustion engine. It is very important to control the frequency of detonations since it is directly proportional to the thrust output of the engine [2]. Once the frequency of engine operation is realized, it will allow proper control on the speed of the aerial vehicle associated with the engine.

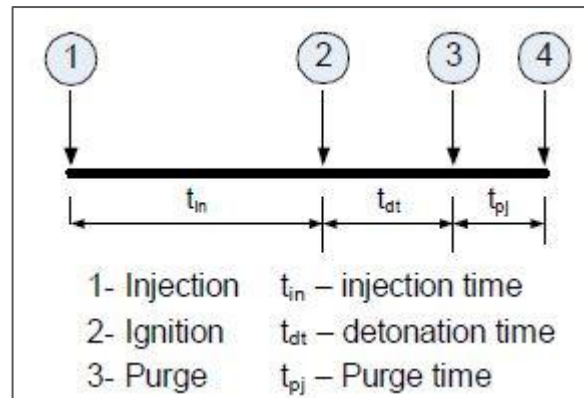


Figure 1.2 : Basic operation cycle for pulse detonation engine [1]

Moreover, the readings from the engine operation need to be taken, stored or viewed in real-time for various purposes such as calibration with analog meter, statistical comparison with computer simulations and feedback response for the control mechanism. Thus, sensors, data processing and data logging need to be integrated besides the control mechanism itself.

To summarize the problem statement, this project focuses on the control mechanism and data logging function integrated with the sensors and actuators associated with FKM's HiREF laboratory Pulse Detonation Engine.

1.3 Project Objective

- a) To design and construct an electronic controller to control a Detonation Engine.
- b) To produce a system which is capable to sample and log the data on the engine operation.

1.4 Scope of Work

The primary scope of this project is to familiarize with the tools (software and hardware) that will be used in this project. The core hardware that will be used is LPCXpresso LPC1769 microcontroller development board while the main software is LPCXpresso IDE (Integrated Development Environment). LPC1769 have many on-board peripherals such as GPIO (General Purpose Input/Output), ADC (Analog Digital Converter) and standard serial communication protocols.

The secondary scope for this project is to design a controller for Pulse Detonation Engine operation. The design of the controller will be based on the operation sequences algorithm required for FKM's High Speed Reacting Flow (HiREF) Laboratory pulse detonation engine. The algorithm will be implemented using LPC1769's GPIO (General Purpose Input Output). It will involve the control of injection, ignition and purge actuators.

Tertiary scope for this project is to establish data communication between microcontroller and computer as well as between microcontroller and external memory storage using standard communication protocol. Analog reading from sensors will be acquired and processed using ADC and LPC1769. The digitized reading from the sensors will either be logged into external memory storage or be displayed and saved using a computer through terminal emulation software.

Overall, the complete system for this project will be able to control detonation engine operations and log all readings into an external memory drive or displayed on computer.

CHAPTER 2

LITERATURE REVIEW

2.1 Pulse Detonation Engine (PDE)

2.1.1 PDE Theory and Concepts

Typical rocket engine uses many concepts for their operation such as deflagration and detonation [5]. Each of these concepts have been tested and applied mostly in aerial vehicle. Deflagration concept was applied in gas turbine engine in an almost constant pressure (isobaric) environment. Each engine was categorized into steady-state and unsteady-state engine [5]. Steady-state engine class examples are turbojets and ramjets which are the most widely used class of engine. Pulse jets and pulse detonation engine were categorized as unsteady-state class engine. Unsteady-state class for those engines were explained by the unsteady condition of the deflagration combustion (pulse jets) and detonation combustion (pulse detonation engine) in pulse nature.

The theory on the Pulse Detonation Engine (PDE) concept comes from two keywords which are detonation and pulse. According to Thomas Bussing and George Pappas in Pulse Detonation Engine Theory and Concepts, “A detonation is a supersonic combustion wave that typically propagates at a few thousand meters per second relative to an unburned fuel/air mixture” [5]. This explained the differences

between typical and detonative combustion in term of power and speed. High-speed nature of detonation leads to an approximation of the process as supersonic shock waves. Pulse nature of PDE was explained by Thomas Bussing and George Pappas as, “rapid detonation process results in constant volume combustion with very high operating frequencies” [5]. From here, pulse can be related closely with frequency or continuous ‘ON’ and ‘OFF’ combustion sequences.

2.1.2 PDE Efficiency

The main argument on why Pulse Detonation Engine was being researched by the aeronautic community was the promising efficiency opportunity. Efficiency is the main concern in any engineering field that involves the use of limited fossil fuel. Efficiency standard was being applied in many applications such as computers, mobile phones, car engine and electric power generation. Efficiency is important to have a high output to cost ratio as well as output to input ratio.

Pulse Detonation Engine differs with conventional engine in many ways which made it more efficient. According to Shmuel Eidelman and Xiaolong Yang, PDE have high structural efficiency due to the absence of turbopumps and compressors in the system [4]. This condition lead to higher thrust to weight ratio than turbo-jet engines [2]. Thus, dramatically reducing the complexity as well as development cost. Shmuel Eidelman and Xiaolong Yang also said that PDE can be developed at low cost using off-the-shelf materials using standard manufacturing methods [4]. Furthermore, the engine can have a thrust variable from 0 to max easily [4]. This can be explained from the controllable pulse (or frequency) parameter of the engine operation. For advance aeronautic application, Philip K. Panicker, Donald R. Wilson and Frank K. Lu said that, “they (PDE) can be used in conjunction with other developing technologies, for example the combustor in scram jet engines can be driven in PDE mode” [2]. In simple, the efficiency factor for PDE was not only on the basis of output-input ratio. PDE are also efficient in structural development and flexibility to integrate into other technology.

2.1.3 Issues On Practical Implementation of PDE

Pulse Detonation Engine needs a very fast detonation pulse to produce enough thrust to drive an aerial vehicle, especially in the Mach speed regime. Philip K. Panicker, Donald R. Wilson and Frank K. Lu reported that there is a source claiming that PDE can operate in Mach number regime from 0 to about 5 [2]. Khalid M. Saqr, Ahmed Faiz, Hassan Kassem, Mohsin Sies and Mazlan A.Wahid also reported that the engine was able to operate until Mach 9 regime [1]. With this promising capability of PDE, the only issue on control mechanism surrounding the practical implementation of PDE is the speed of actuators where it is needed to perform the fuel/oxidizer injection, ignition, and purge phase of the engine's operation.

Practically, the actuator was divided into mechanical and electrical valves. The fastest mechanical valve was currently used in F1 racing car engines which can operate at 20,000 rpm [2]. To achieve that performance, such system has a rather complex system contributed by many moving parts involved. Unlike mechanical valve, solenoid (electrical) controlled valves have simpler build which reduces the complexity. They are also very fast reacting where it can have a fraction of millisecond (ms) reaction time. Furthermore, solenoid valves can be controlled precisely using TTL (Transistor-Transistor Logic) signal from a computer [2]. In addition, such system is commercially available off-the-shelf today following the increasing demand in electronic injection for car engine. Thus, electronic control mechanism is more favored for PDE application.

2.2 Rocket Engine Control System

The Space Shuttle Orbiter Main Engines provide the primary thrust for NASA (National Aeronautics and Space Administration) Orbiter vehicles. The engine has an engine-mounted electronic control system called the controller [3]. The controller has the capability to do self-checkout prior to flight and perform monitoring and reporting of engine status and condition during all phases of

operation [3]. The phases of operation are engine start, main stage and shutdown. All of these phases are controlled through an electronic controller. As reported by P. F. Seitz and R. F. Searle, the controller will receive command inputs for various operational phase, positions appropriate valves, monitors the engine for required performance, precisions and conditions, and provide redundancy management [3]. As a result, Space Shuttle Main Engine (SSME) controller will vary the thrust output and mixture ratio of fuel/oxidizer in the mixing chamber with respect to command input. To execute such command, the controller will transmits signals for positioning of the valves actuators, switching hydraulic and pneumatic solenoid valves and controlling the spark ignition [3].

In order to have maximum precisions and accuracy for the performance and desired SSME conditions, a close-loop feedback control system was implemented [3]. This approach can be applied in any control method to achieve certain accuracy at output. The system will continuously monitor the controlled parameter and compare it with the command value and gives appropriate signals to perform calibration measures. An example to a close-loop feedback system is depicted in Figure 2.1 for SSME thrust control system below. With such feedback system, the calibration process is more economical where there is no need for resizing and additional machining on the engine's structure in order to achieve desired operating condition. In addition, the system is well established and proven many times for military flight control system [3].

The technology is progressing very fast and sometimes there is certain new control configuration needed by the SSME. New configurations may be required to adhere to certain calibration standard or operation sequences. To perform rapid changes on how the controller operates the engine, a feature called programmable digital logic is implemented [3]. With this implementation, any changes in operational sequences and function can be made by modifying the computer software installed. This feature is almost similar to the main feature of any microcontroller where the software can be changed by direct programming through computer software. Thus, reducing the time and cost for hardware redesign [3].

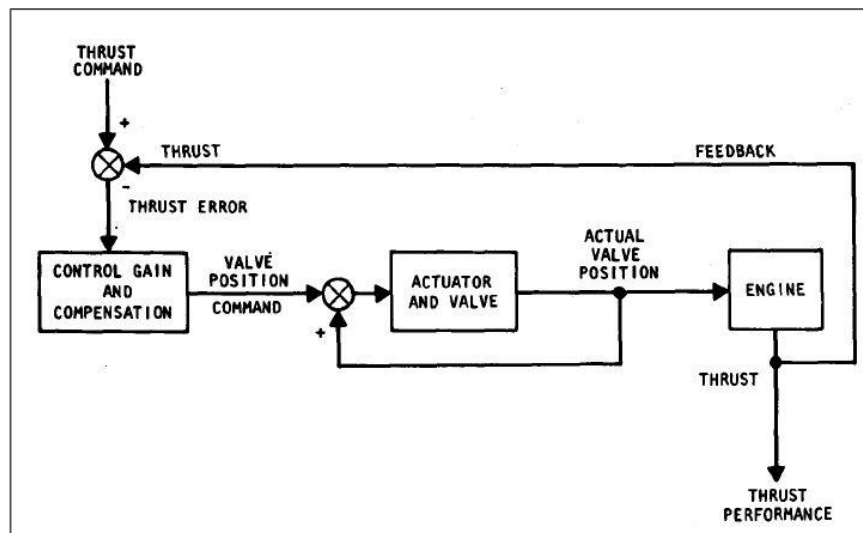


Figure 2.1 : SSME thrust control system (with close-loop Feedback) [3]

2.3 FKM's HiREF Laboratory Pulse Detonation Engine

Universiti Teknologi Malaysia (UTM) has developed a Pulse Detonation Engine prototype. The research-purpose prototype was built and tested in High Speed Reacting Flow (HiREF) laboratory in the Faculty of Mechanical Engineering (FKM) under direct funding from the Malaysian Government. Development of HiREF's PDE includes the design and development of PDE fuel admission, detonation tube, control, and data acquisition systems [1]. Figure 2.2 shows the actual developed PDE by HiREF.

2.3.1 Pulse Detonation Engine Structure

The Pulse Detonation Engine structure consists of fuel-oxidizer, ignition and purging actuators. It operates on propane and oxygen with single detonation tube [1]. It was designed to operate up to 100Hz detonation cycles which consist of four stages as depicted in Figure 1.2 previously. The mixing chamber has a diameter of 50 mm and 142 mm long while the stainless steel detonation tube has an inner diameter of 50 mm and 600 mm long [12]. The oxygen and propane are injected through a

total of 6 injectors (four for oxidizer, two for fuel). For this engine, Compressed Natural Gas (CNG) injectors based on electrically controlled solenoid valve as in Figure 2.3 were used for the fuel-oxidizer and purge air admission system. From Figure 2.3, the left side picture is the CNG injector with while the right side is the mounting structure built to interface the injectors with PDE.

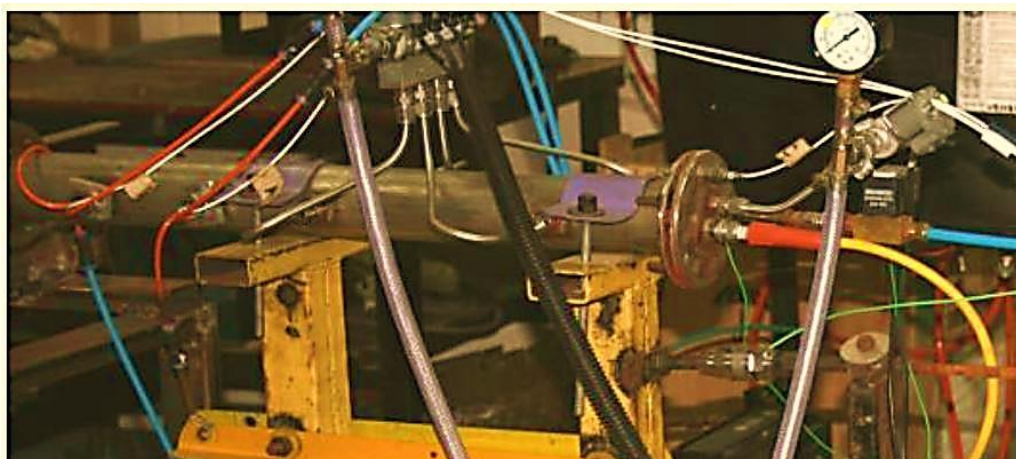


Figure 2.2 : Actual PDE prototype developed by HiREF [12]

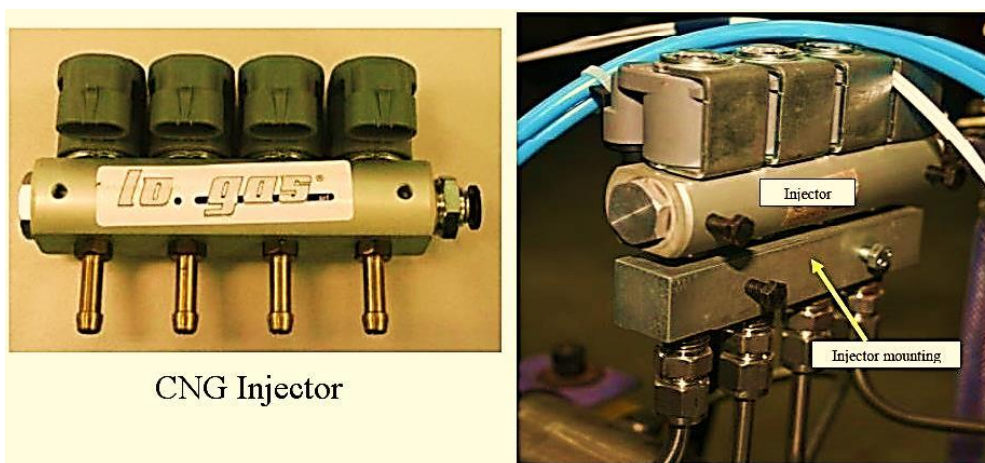


Figure 2.3 : CNG injector from LO-Gas [12]

2.3.2 Pulse Detonation Engine Operation

The operation framework for FKM's HiREF Pulse detonation Engine is based on Figure 1.2 where it is a cycle that comprises of fuel-oxidizer injection, ignition and purge stages. A precise time interval between each stage is needed. From a Faculty of Mechanical Engineering (FKM) master thesis entitled "Development of Pulse Detonation engine", the time interval between fuel-oxidizer injection and ignition is 30% of the whole cycle. The same percentage also applies to the time interval between ignition and purge stage. Last of the whole cycle, time interval between purging and fuels-oxidizer reinjection is 40% [12]. All of the time interval proportions can be visualized as in Figure 2.4.

Fuel-oxidizer injection	Ignition (detonation)	Purge (removes burnt fuel)
30%	30%	40%

Figure 2.4 : Time interval between each stage

2.3.3 Pulse Detonation Engine Control Circuit

The control circuit for FKM's HiREF PDE is divided based on each actuator which is fuel-oxidizer injection, ignition and purge air circuit. As described before in section 2.3.1, the actuator for fuel-oxidizer and purge air injection is CNG injectors based on electrically controlled solenoid valves. The control circuit for ignition is based on an automotive spark plug [12].

The solenoid valves (CNG injectors) for fuel-oxidizer and purge air injection have 12V and 24V rating respectively [12]. The difference in voltage rating was probably due to the difference in maximum pressure rating for fuel-oxidizer and purge air where purge air injection needs higher pressure than fuel-oxidizer injection.

The ignition control circuit based on automotive spark plug have special circuitry that involves a very large capacitor and power supply circuit. It can be understood further as visualized in Figure 2.5. The spark plug provides ignition when

the power supply is disconnected to allow capacitor discharges current [12]. Thus, it can be understood that the control signal for ignition circuit is active-low.

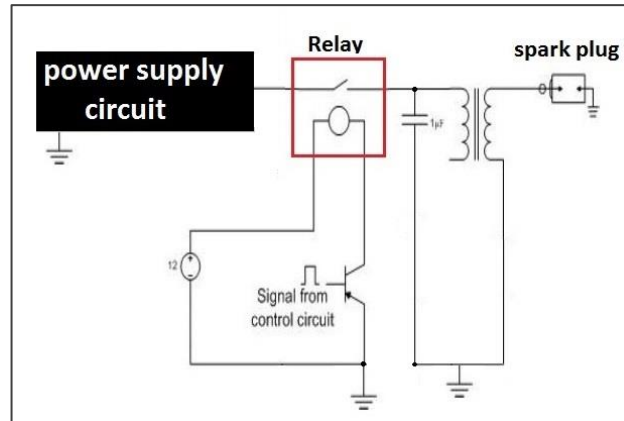


Figure 2.5 : Ignition circuit

2.3.4 Measurement Sensors

The sensors for HiREF's pulse detonation engine are used to measure detonation pressure, detonation acceleration and the force produced from the fuel-oxidizer detonation [12]. All these measurements require three different transducers which are mounted on the PDE. They are KISTLER 211B300 PIEZOTRON pressure transducer for measuring pressure, KISTLER 8704B5000 accelerometer for measuring acceleration and KISTLER 9331B load cell for measuring force. These transducers will generate output voltage with respect to the parameters it senses.

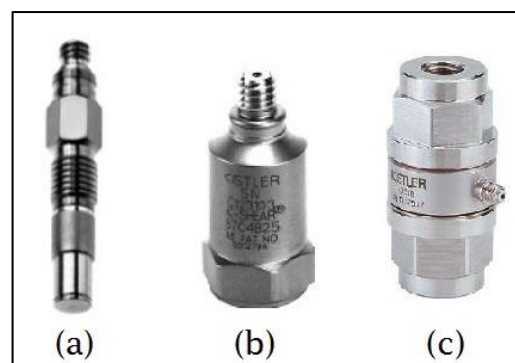


Figure 2.6 : (a) Pressure transducer, (b) accelerometer, (c) load cell

2.4 LPC1769 Microcontroller Unit (MCU)

LPC1769 is a microcontroller unit (MCU) manufactured by NXP Semiconductors. It is based on widely used industrial standard 32-bit ARM Cortex-M3 architecture [14]. LPC1769 was intended for applications such as electronic-based metering, alarm systems, lighting, industrial networking and motor control. Based on LPC17xx microcontroller series user manual, the list below is some of the peripherals included on-board LPC1769 [6].

- 3.3V microcontroller system.
- Up to 120 MHz CPU frequency.
- 512 kB flash memory (program/firmware), up to 64 kB data memory (RAM).
- 4 UARTs
- 2 CAN (Controller Area Network) channel.
- SPI interface.
- 3 I²C interfaces.
- 8 channel 12-bit ADC (Analog-Digital Converter).
- 10-bit DAC (Digital-Analog Converter)
- 4 general purpose timers.
- 6-output general purpose PWM (Pulse Width Modulation).
- 1 USB interface (full speed class)
- 70 general purpose I/O pins.
- Support both standard JTAG and ARM Serial Wire Debug (SWD) interface.

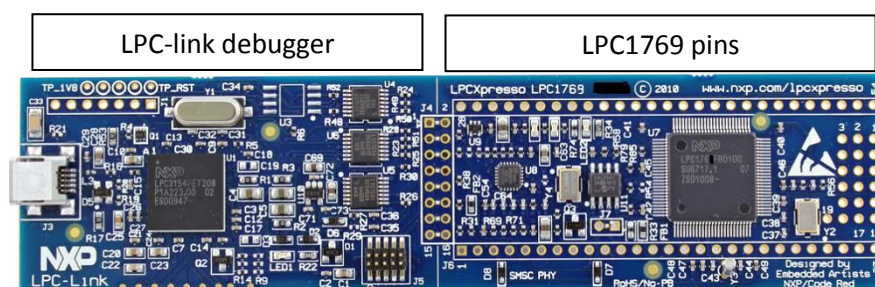


Figure 2.7 : LPCXpresso LPC1769 development board

2.4.1 LPCXpresso LPC1769 Development Board

LPCXpresso LPC1769 is the core development platform for LPC1769 in the market today. It is a development board by Embedded Artist based on NXP Semiconductor's LPC1769 microcontroller unit (MCU) [17]. Figure 2.7 shows the LPCXpresso LPC1769. This development board can be fitted on breadboard for testing and fast embedded system prototyping.

2.4.2 LPCXpresso IDE (Integrated Development Environment)

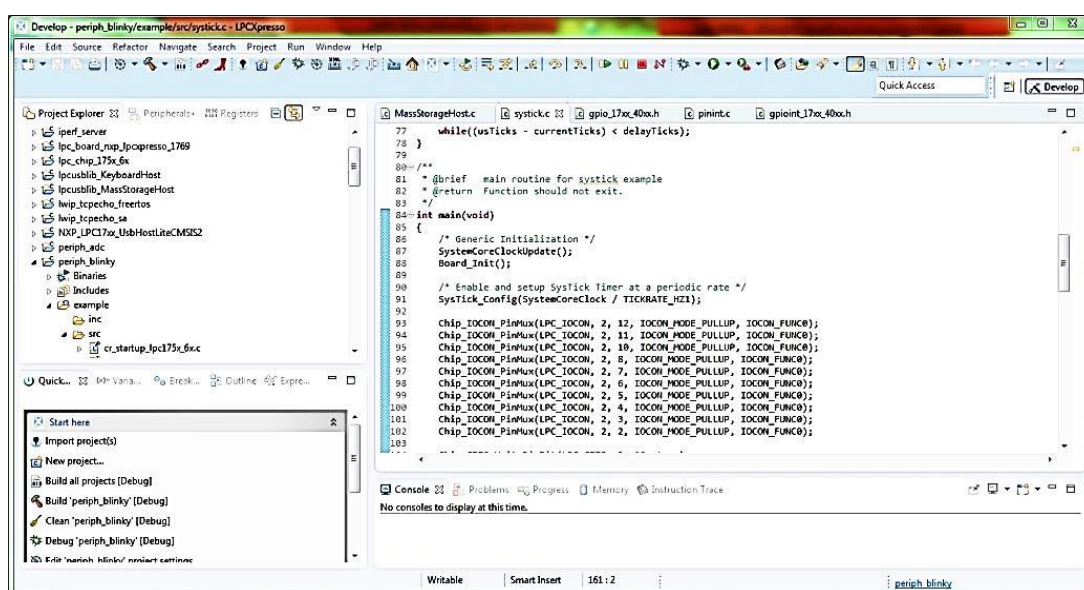


Figure 2.8 : LPCXpresso IDE workspace

LPCXpresso LPC1769 is supported by LPCXpresso IDE for embedded system development and debugging. It is a development environment based on Eclipse IDE for NXP's ARM-based microcontrollers. With this IDE, NXP Semiconductors through its developer's forum website, *LPCware.com* provide access for example libraries and standard peripheral libraries [15, 18]. Figure 2.8 shows the workspace and GUI for LPCXpresso IDE.

All peripheral libraries for LPC17xx microcontroller series are being supported by NXP Semiconductors until today. They can be downloaded online or found in the example directories of *LPCXpresso* IDE. The libraries were a starting

point for any embedded system developers who are using NXP's LPC microcontroller as their implementation platform.

2.5 Serial Communication

Any microcontroller applications need a medium for interaction with its user. The user may interact with microcontroller for control, debug, program or display purpose. The most fundamental function of communication medium is for debugging purpose. Debugging enables user to observe the behavior of a firmware/program when testing microcontroller applications. For all these purposes, embedded system developers need to establish communication protocol such as serial communication.

Serial communication is the most widely used protocol for microcontrollers today. Some serial communication protocols include Ethernet, CAN, SPI, USB, I²C, RS232 and UART/USART. The most common serial communication protocol for today's microcontrollers is by using RS232 or UART/USART. LPC1769 microcontroller has 4 sets of UART pin interfaces. Each UART interfaces have a pair of transmit (TX) and receive (RX) channels.

2.5.1 UART Interface on Microcontroller

UART interface on microcontroller is necessary for UART serial communication before sending data to computer through RS232 cable. LPC1769 microcontroller which will be used in this project is 3.3V microcontroller system. That means the logic High and Low for this microcontroller will be at 3.3V and 0V. This situation requires transceiver or logic converter to communicate with different system such as between microcontroller and USB port as well as between microcontroller and RS232 port.

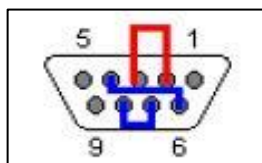


Figure 2.9 : RS232 Loopback Testing (shorting red coloured pins)

2.5.2 Serial Communication Interface Using Computer

RS232 cable interfacing for RS232 communication protocol with earlier computer model is made possible with RS232 port on-board. Thus, the connection is tested by performing loopback testing using pre-installed *HyperTerminal* windows-based terminal emulation software on computer. It was done by shorting receiver (RX) and transmitter (TX) pins which are pin 2 and 3 of DE-9 socket as shown in Figure 2.9. The data input such as from keyboard will be shown in *HyperTerminal* window, proving there is a connection between the PC itself (loopback) through RS232 cable.

For most computers and laptops today, RS232 port was not provided by the manufacturers. The only available and most common communication ports for today's computers and laptops are USB, LAN and WLAN. There are many USB-UART converter models available in the market to solve such problem. Typically, the cable for the converter consists of a power supply (VCC), ground (GND), transmitter (TX) and receiver (RX). Simply short the RX and TX cable to test the connection of the converter and observe the input from keyboard appear in the *HyperTerminal* window.

CHAPTER 3

PROJECT METHODOLOGY

The project methodology was done based on Universal Design Methodology (UDM) in Appendix 1 taken from <http://www.embedded.com>. The methodology can be applied to almost any projects with various engineering background which requires hardware realization. Based on Appendix 1, a project methodology shown in Figure 3.1 was developed to specifically parallel with the design aspects of this project.

3.1 Project Methodology Flow Chart

Based on the flow chart in Figure 3.1, the project was started with reviewing literatures from related journals, thesis, proceedings and previous works. By reviewing the literatures, some detail aspect in the project such as definitions, history, theories and practical solution examples can be found and studied thoroughly. Thus, problem statement and scope of the project may come afterwards.

Next, the work was continued by determining and familiarizing the tools and softwares needed by the project. This stage may involve trainings or hands-on tutorials on the related tools and softwares. Then, the project was proceed to the design phase of the project where the algorithm of the PDE operation was developed. After that comes the testing/verification and discussion/conclusion phase. The

project will keep on repeating the testing/verification again and again until the desired results are achieved.

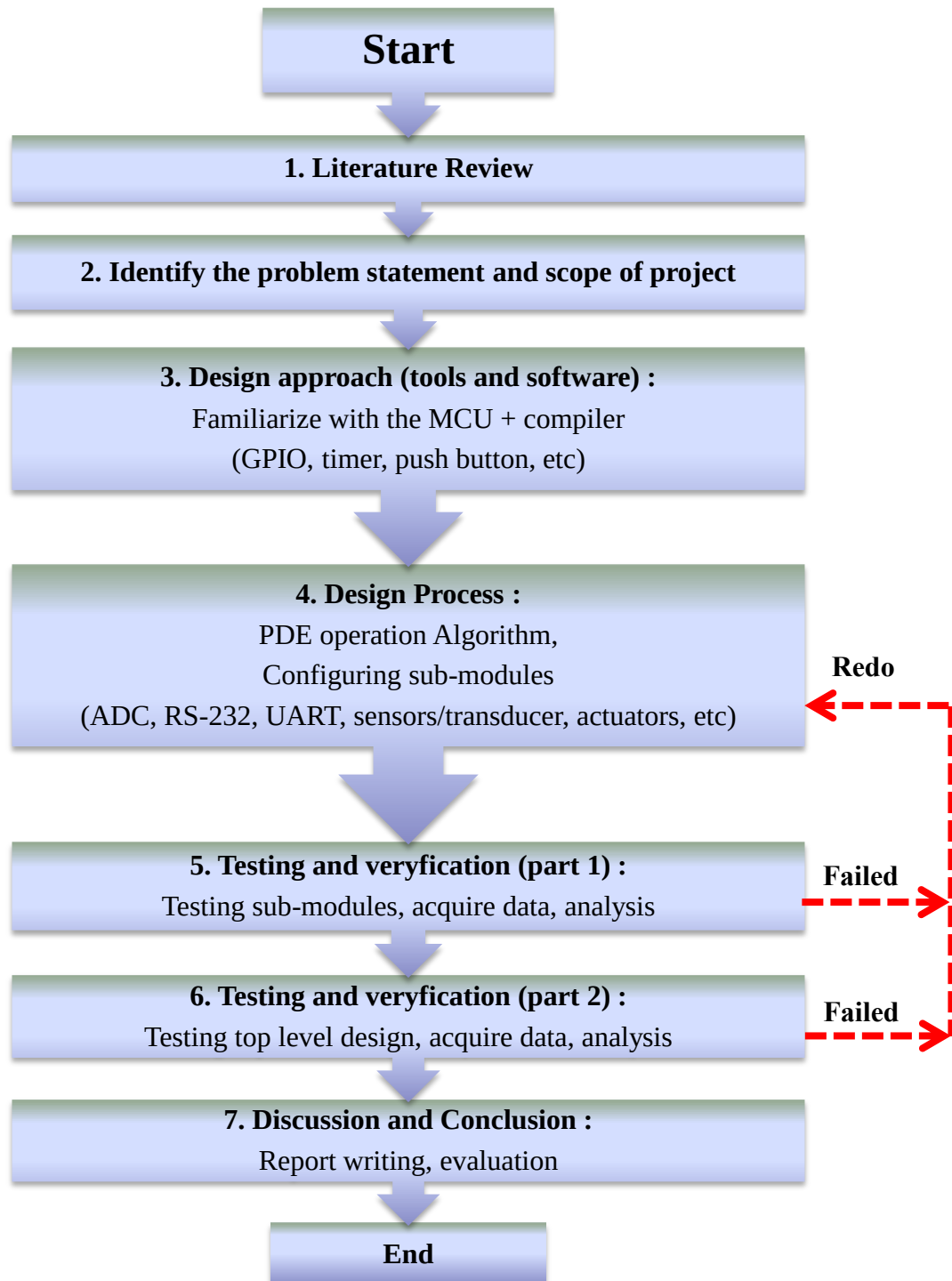


Figure 3.1 : Core project methodology

3.2 Top Level Design

In order to execute with the project, a rough view of the actual final prototype is visualized as in Figure 3.1. Based on the figure, the LPCXpresso LPC1769 development board are interfaced with sensors or transducer, actuators, external memory and communication line. The sensors will be interfaced using ADC (Analog-Digital Converter) pins while the actuators with GPIOs. For external memory and communication, they will be interfaced with communication peripheral pins provided in LPCXpresso LPC1769 development board such as USB, SPI and UART pins as these pins are very specific with its functions. From the communication interface, the system can be connected to computer for debugging, controlling and displaying purpose.

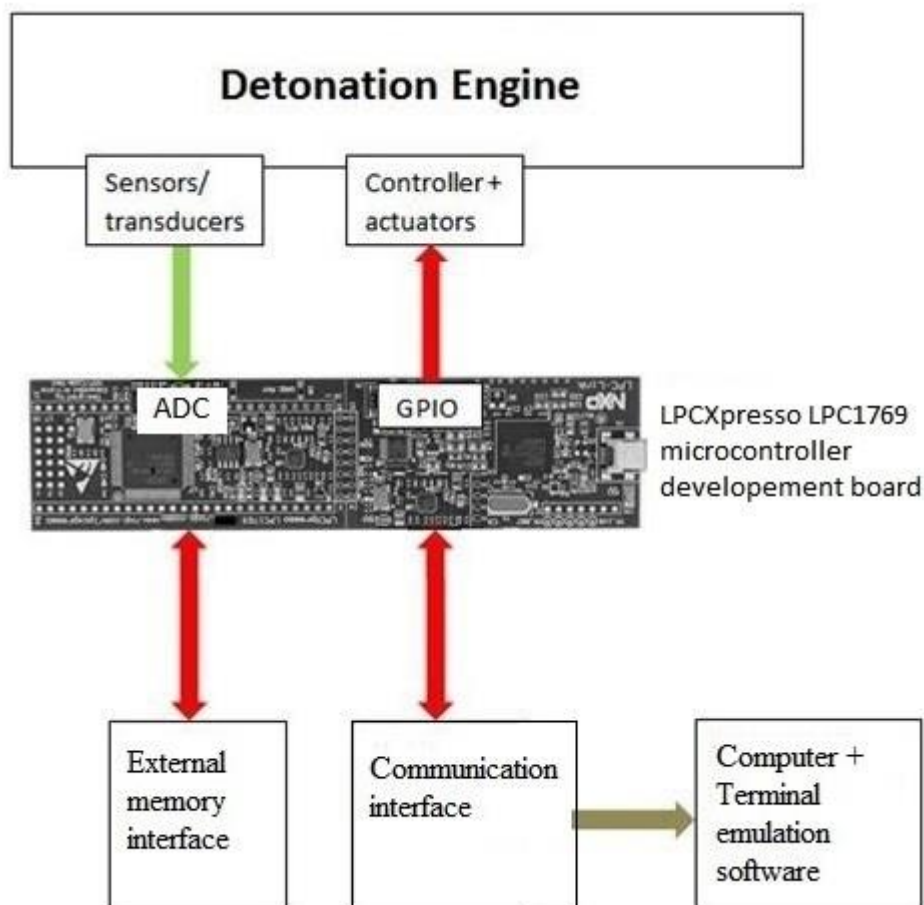


Figure 3.2 : Top level graphical representation of the project

3.3 Serial Communication Interface

Serial communication enables user to display data such as ADC sampling value in real-time using terminal emulation software such as *HyperTerminal* or *TeraTerm* through a computer. Furthermore, the software can be used to provide simple GUI for the user to operate any applications that are using microcontroller as their platform.

Most manufacturers nowadays are moving to USB standard for most peripheral interfaces such as mouse, keyboard and external memory while keeping the interface for VGA and Ethernet ports. These technological developments lead to some constraints for embedded system developers in their design. In order to counter such problem, USB-UART converter is implemented. There are many manufacturers for the converter as well as the transceiver IC.

3.3.1 UART-USB Converter

For this project, UC00B V2011 from Cytron Technologies Sdn Bhd is used. This implementation makes the project easier to be debugged and at the same time providing serial communication directly from the microcontroller itself without having to build any special interfacing circuitry. More than that, the USB-UART converter is powered by the USB host at which it is connected where for this project, it is the computer. It has dual voltage rating for 5V and 3.3V microcontroller system that can be easily switched using a jumper pin. Thus, serial communication interface using USB-UART converter is the best communication method for this project. Figure 3.3 below shows the converter in actual form.

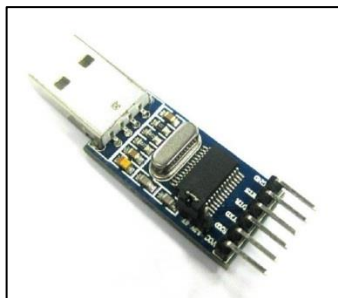


Figure 3.3 : UC00B V2011, Cytron's USB-UART converter [13]

3.4 Pulse Detonation Engine Controller Design

The PDE controller is divided into hardware and firmware module. The hardware module consists of microcontroller GPIO pins, current reducing resistors and optocouplers (firmware module will be discussed in section 5.1.2). Based on FKM's laboratory (HiREF), their Pulse Detonation structure consist of several injectors and a relay to be driven using microcontroller. A total of 8 control signal was needed for the controller module. Table 3.1 shows the tabulation of solenoid valves and relays for the engine.

Function	Actuator	Number of actuator	Control input signal (Voltage, VDC)	Detail
Fuel-oxidizer injection	Solenoid valve	6	12	Active high
Ignition	Relay	1	12	Active low
Purging	Solenoid valve	1	24	Active high

Table 3.1 : Tabulation of actuators and control voltage

The controller is functioning by sending logic signals through GPIO pins of *LPCXpresso* LPC1769. Theoretically, the signal from GPIO pins should drive the relay for ignition circuit and solenoid valves for fuel-oxidizer and purge injectors. Unfortunately, the microcontroller (LPC1769) can only give a maximum of 3.3V output from its GPIOs and limited current [6]. Thus, an isolator circuit need to be

built between microcontroller and associate actuators. For this purpose, isolator circuit based on optocoupler was designed.

3.4.1 Isolator Circuit

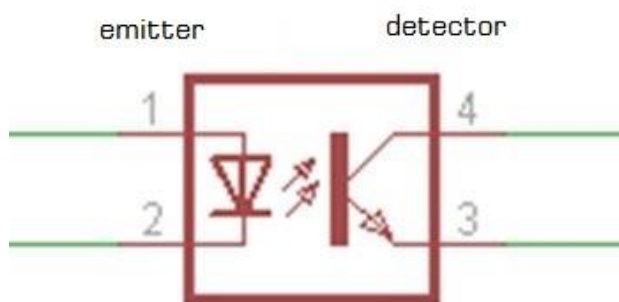


Figure 3.4 : Optocoupler internal schematic

An optocoupler comprises of an emitter and a detector. Based on figure 3.4, there is no physical connection between them which made an optocoupler an isolator between low and high voltage/current side. When the emitter side is connected to a voltage, an infra-red wave will be generated from emitter's diode and trigger the Base terminal of detector's transistor and thus, connecting the Collector (pin 4) and Emitter (pin 3). In other words, the emitter side can be connected to the microcontroller GPIOs while the detector side can be connected to a high voltage/current of a separate power supply where the high voltage/current side will have high voltage/current whenever there is an infra-red trigger from the emitter. Figure 3.5 shows the schematic of connections between microcontroller (LPCXpresso LPC1769) and output of optocoupler. The optocoupler model being used is 4N25 (black package type). This optocoupler-based control mechanism is suitable for high speed switching between logic Low and High signals from microcontroller's GPIOs [9].

The output voltage from optocoupler depends entirely on input voltage, current into emitter's diode and power supply voltage at detector. Protection for the emitter input current is also need to be considered using a current limiting resistor at

each GPIO pins. Table 3.2 shows the tabulation of resistors and supply voltage values.

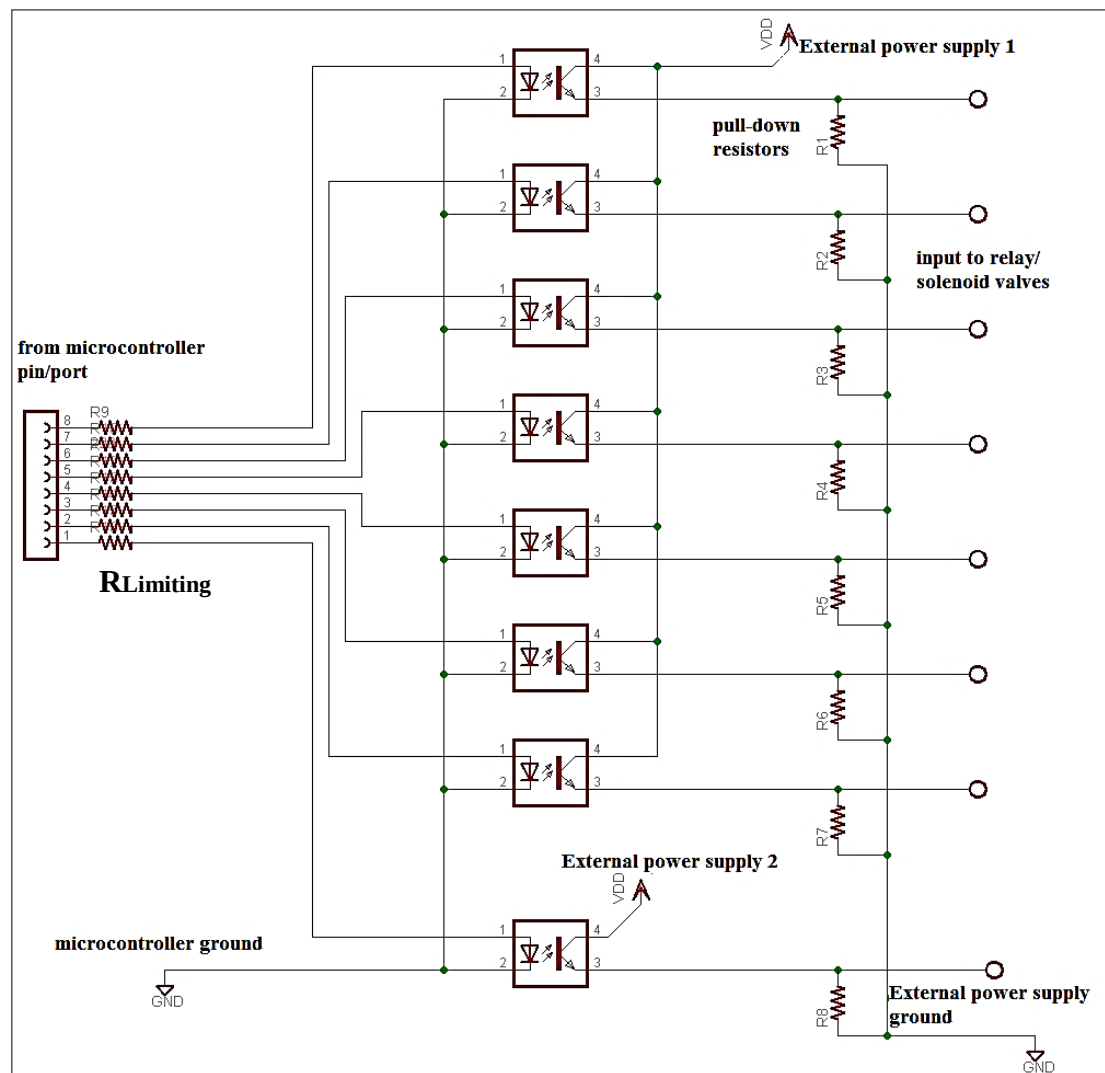


Figure 3.5 : Isolator circuit schematic (for all 8 control pins)

Components/ Supply Voltage	Value
Microcontroller pin (High, Low)	3.3 V, 0 V
$R_{Limiting}$	220 ohm
External power supply 1	12 V
External power supply 2	24 V
Pull-down resistor	3.6k ohm

Table 3.2 : Resistor and supply voltage value for isolator circuit

3.4.2 PDE Controller Operation

From section 2.3.2 previously, the time interval between each stage is proportioned precisely at 30%, 30% and 40% for fuel-oxidizer injection, ignition and purge stage respectively. If Pulse Detonation Engine operation is at 1 Hz, the time interval required is 300 ms, 300 ms and 400 ms respectively. This configuration can be changed accordingly using simple mathematical calculation as shown in example below. Table 3.3 shows timing interval examples according to frequency of PDE operation. Even though the timing interval is proportionate according to the specific percentage proportion as mentioned before, it can be changed by the user as long as the timing interval is correctly configured using the same mathematical calculation below.

Timing proportion (factor) calculation:

- Fuel-oxidizer injection = 30% or $30/100 = 0.3$ factor.
- Ignition = 30% or $30/100 = 0.3$ factor.
- Purge = 40% or $40/100 = 0.4$ factor.

Timing interval calculation:

Frequency = **2** Hz,

Time required for 1 cycle = $(1/\text{frequency}) \times 1000$ ms

= **500** ms

Fuel-oxidizer injection timing = (factor) $\times$ (time required for 1 cycle)

= 0.3×500 ms

= **150** ms

Ignition timing = (factor) $\times$ (time required for 1 cycle)

= 0.3×500 ms

= **150** ms

Purge timing = (factor) X (time required for 1 cycle)

= 0.4 X 500 ms

= **200 ms**

Frequency (Hz)	Timing Interval (ms)			Time Required for 1 cycle (ms)
	Fuel-oxidizer injection	Ignition	Purge	
1	300	300	400	1000
2	150	150	200	500
5	60	60	80	200
8	37.5	37.5	50	125
10	30	30	40	100
12	25	25	33.33	83.33

Table 3.3 : Example timing intervals with respect to PDE operation frequency

3.4.3 PDE Controller Mechanism & Firmware Design

The mechanism of PDE controller is not simply providing logic High or Low operation to the solenoid valve injectors and ignition relay. The switching of logic High and Low for PDE controller need to perform at a very precise and accurate timing proportion from fuel-oxidizer injection stage to ignition and purging stage in one complete cycle repeatedly as in Figure 2.4.

LPC1769's timer peripheral was used to provide timing control as well as accurate delay between each stage. Thus, the firmware module for PDE controller is comprises of GPIO and internal timer only. A flow chart as in Figure 3.6 was constructed as a framework of the code for microcontroller (LPC1769).

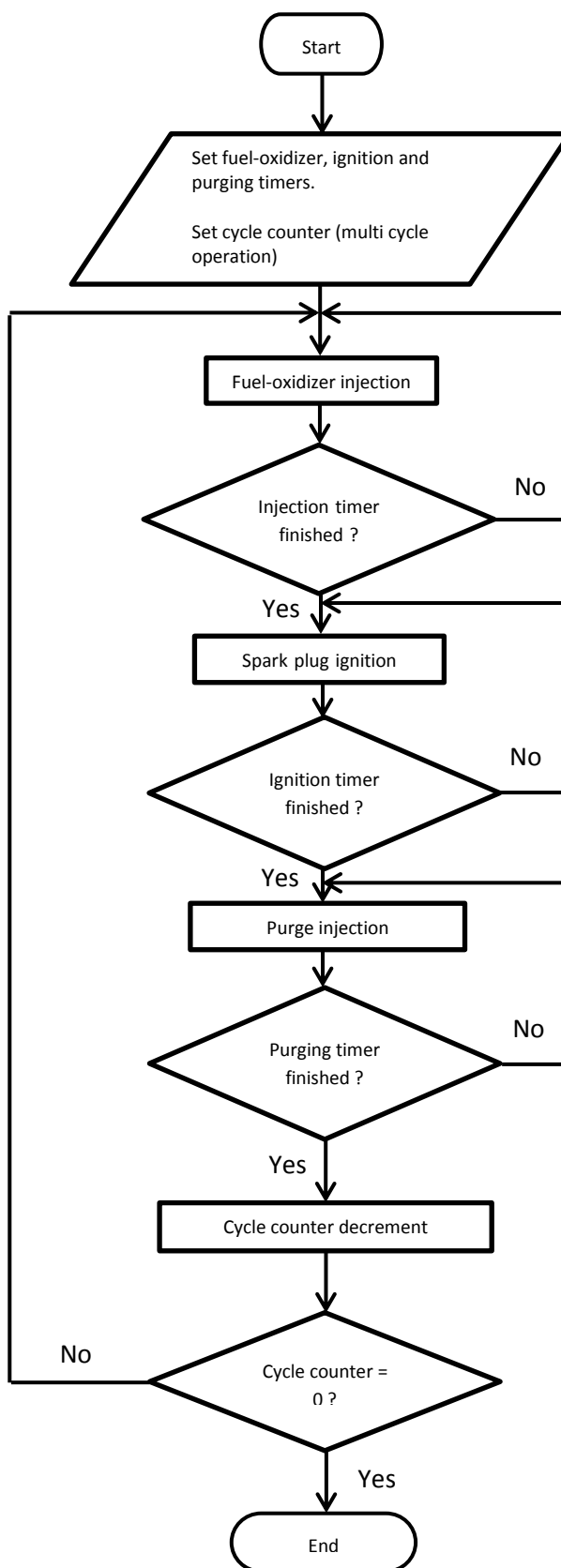


Figure 3.6 : PDE controller flow chart

3.5 ADC Data Logging

From the objective, data samples from ADC need to be stored for PDE analysis. The data logging hardware module consist of ADC on-board LPC1769 microcontroller, sensor (pressure transducer) mounted on the PDE tube, external memory storage and serial communication interface. ADC for LPC1769 has 12-bit capability at 200kb/s conversion speed [6]. Since LPC1769 is a 3.3V microcontroller system, the resolution of the ADC is 0.8mV [11].

3.5.1 ADC Signal Attenuator

Some considerations need to be made when interfacing the pressure sensor with ADC input pins. The voltage rating for the sensor must not be outside ADC input's absolute maximum rating. The sensitivity of the transducer is 1.086mV/psi. The maximum allowable pressure is 5000 psi. Thus, the sensor supposedly will generate a maximum voltage of 5.43V. The maximum input voltage for LPC1769's ADC is 3.0V while the maximum output voltage for the sensor was 5.43V [11]. Attenuation process needs to be made to the sensor's output before it is being fed into ADC pins. A maximum voltage of 5.5V is assumed to simplify design parameter. Thus, two 5.5V to 3.0V attenuator configurations was designed.

The first configuration were made based on typical inverting amplifier which is cascaded into two stages to provide zero degree phase shift on the signal since single amplifier will produce inverted signal phase. Figure 3.7 depicted the attenuator's configuration. The second configuration was based on a buffer using op-amp (LM741) and voltage divider circuit as in Figure 3.8. The result for these attenuator analysis will be discussed in chapter 4.

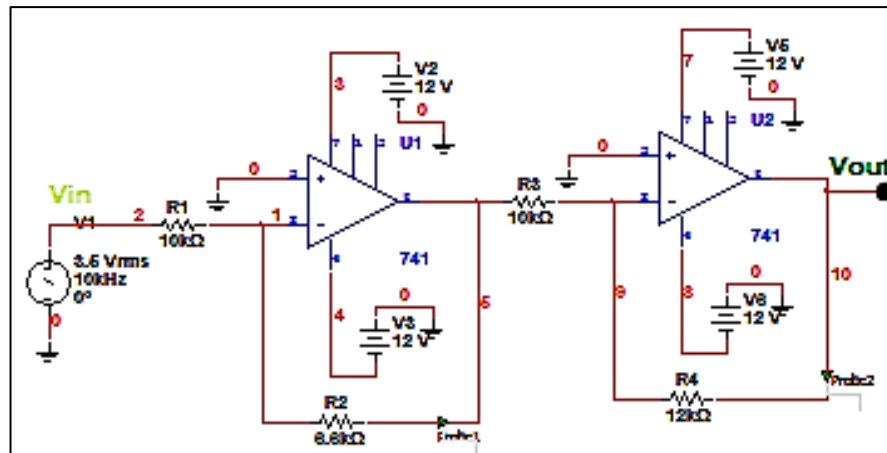


Figure 3.7 : Attenuator (Configuration 1)

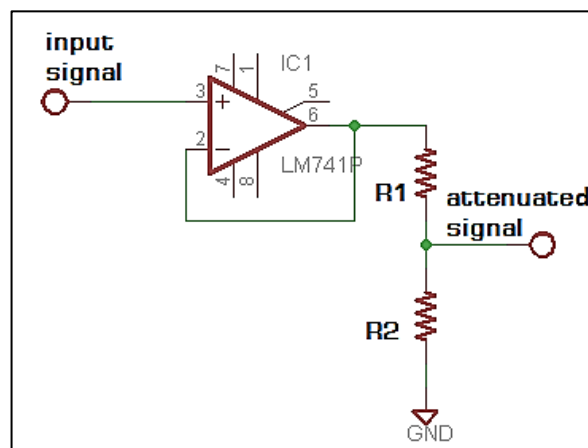


Figure 3.8 : Attenuator (Configuration 2)

3.5.2 External Memory Storage

The readings from ADC input needs to be stored by means of a memory storage device. Due to the research purpose of Pulse Detonation Engine, it is very critical that the data on the engine's operation need to be sampled and stored for further research analysis. The memory device needs to have easy access using most computers today. Thus, USB mass storage was the most suitable medium for ADC data storage.

NXP's LPC product range is very popular with USB-based driver libraries. In their developer's forum website, www.LPCware.com, there is a special section for USB peripheral libraries for NXP's LPC product lines. NXP provides up-to-date

USB peripheral drivers for LPC microcontrollers such as USB Host, HID, CDC and Mass Storage applications. These drivers had been supported by NXP for years.

Unlike other model, the easiest way to have an external mass storage device for LPC1769 microcontroller is through USB peripheral on-board. It has one USB peripheral with a maximum of 12Mb/s (USB 2.0 full speed class) data exchange rate. In order to use USB flash memory, the microcontroller needs to function as a host to the USB flash memory device. Figure 3.9 shows the implementation of USB Host mode using LPC1769 while Figure 3.10 shows the actual USB Host implementation using breadboard.

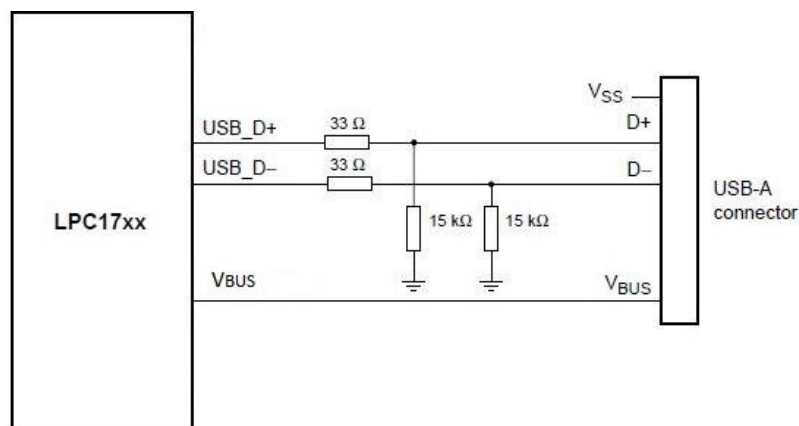


Figure 3.9 : USB Host implementation for LPC1769 [10].

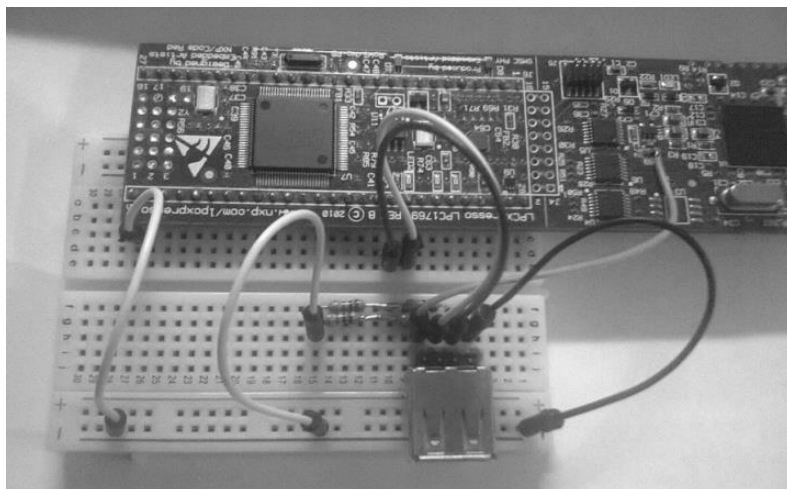


Figure 3.10 : USB Host actual implementation (using breadboard)

3.5.3 ADC Data Logging Firmware

ADC data logging firmware consist of several parts which are ADC, timer, USB Host and FAT File System. As previously explained in earlier chapters, ADC was used to convert analog signal from pressure transducer to digital form. Based on flow chart in figure 3.11, the micro second timer will start first after system initialization. Then, the system will detect whether USB mass storage device was connected or not. If it was detected, the system will proceed to start the ADC peripheral and logged ADC value into the USB memory storage while the end flag is false. The end flag was triggered externally, providing data logging control for user to stop the data logging.

The data logging from LPC1769 to USB flash memory was done using Chan's FAT File System library. This library is specialized for used in embedded system applications. The source code for Chan's FAT File System is open source and being utilized by NXP Semiconductor to provide example source code library for USB mass storage application. FAT (File Access Table) File System is a computer file system architecture. In simple, FAT File System is a firmware interfacing medium for accessing standard memory storage devices such as USB flash memory, SD card and hard disc drive.

Regardless of any data logging configuration, the USB mass storage device was detected by a function from Chan's FAT File System library. From flow chart in figure 3.11, after ADC value was put into a buffer, data from the buffer along with micro second timer value and counter value are converted into a character string using C language function, `sprintf`. From here, the data logging operation was controlled using functions from Chan's FAT File System library. The functions are sequenced in a specific manner to ensure the data in a created text file in USB mass storage is updated each time the data was logged, instead of creating new file or overwriting the file created.. Table 3.4 shows the sequence of functions to logged ADC values into USB mass storage.

Sequence	Function	Comment/detail
1	f_mount	- Register/Unregister a work area
2	f_open	- Open/Create a file
3	f_lseek	- Move read/write pointer, Expand file size
4	f_write	- Write file
5	f_close	- Close an open file

Table 3.4 : Chan's FAT File System function in sequence.

From Table 3.4, the f_mount function will detect the connection of USB mass storage device and register a file pointer. If there is no problem, the firmware will proceed to ADC data sampling. After the data was put into a buffer and converted to a string along with micro second timer and writing counter, f_open will open the file for any operation. This function can be configured into many modes such as creating new file and opening existing file by specifying specific figures or definition in the arguments. After that, f_lseek will move the read or write pointer of the file. Then, f_write writes the data specified by a buffer pointer in its arguments. F_lseek function ensures the data are written by f_write function at the end of existing data in the file. If this function is not used, the data will be written at the first address. Thus, overwrites the existing data. At the end of logging operation, f_close will close the opened file. Inability to do so may cause problems to the file opened for future logging operation. The whole operation was repeated each time the ADC conversion was finished while the end flag is false.

The ADC peripheral example library provided by NXP Semiconductors can only perform simple ADC sampling operation and displayed through UART serial interface. Major adjustment was made to include micro second timer and USB Host peripherals in the same system. The stamping for micro second timer is configured using SysTick timer interrupt peripheral on-board LPC1769 microcontroller. The smallest time interval for this timer is 1 micro second.

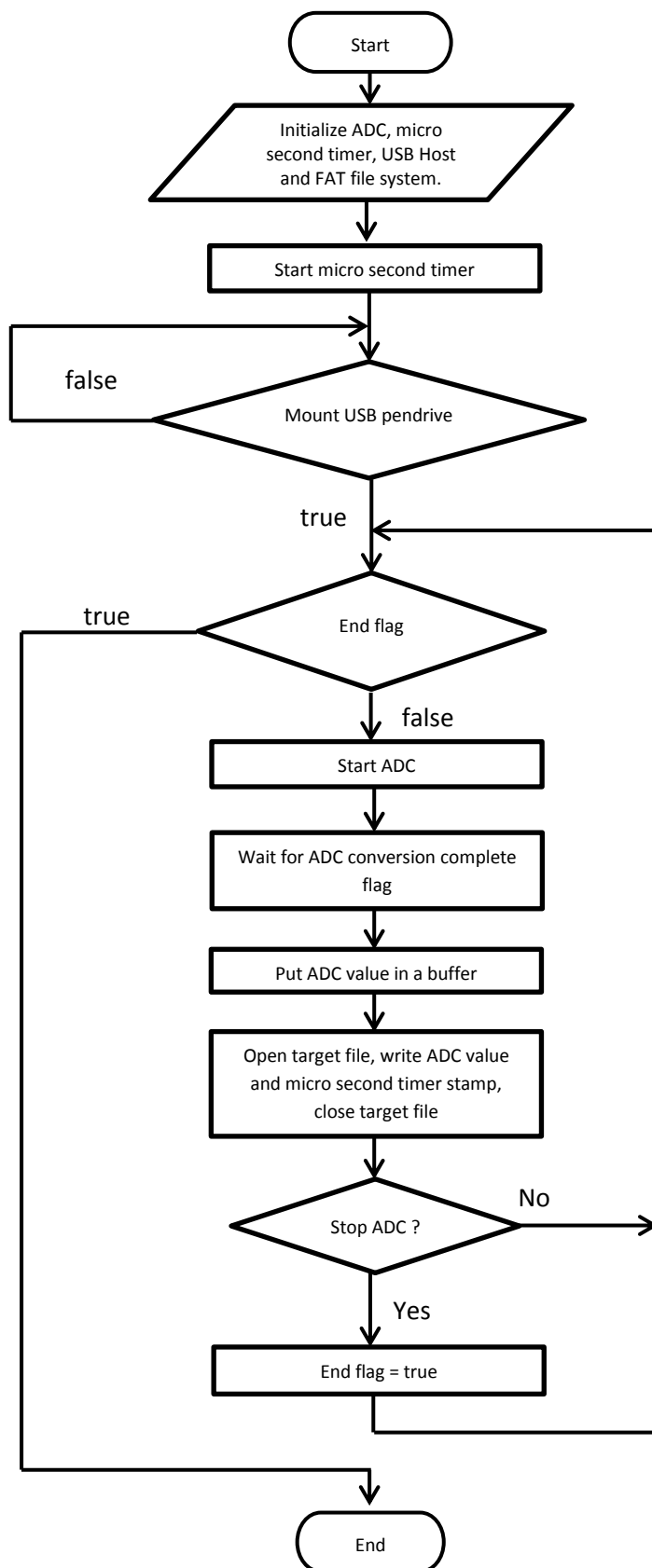


Figure 3.11 : ADC data logging operation flowchart

3.6 Testing and Verification

Testing and verification need to be done to ensure the expected functionality of all devices or application is met. For this project, the test was divided into two segments which are PDE controller functionality and ADC data logging performance.

3.6.1 PDE Controller Testing and Verification

The controller circuit was tested using light emitting diode (LED) as an indicator for high speed logic High and Low switching of PDE controller pinouts. This testing method is used to ease the testing environment for PDE controller. This is due to the fact that the FKM's HiREF laboratory was under maintenance and pulse detonation. Moreover, the pulse detonation engine in the laboratory was currently being upgraded to be used in power generation research using PDE as core platform. Figure 3.12 and 3.13 shows the test circuit schematic and actual test circuit respectively.

From figure 3.12 and 3.13, there are a total of 8 LED which represent the control pins for all actuators. From figure 3.13, we can assume that LED 1 (most left) until LED 6 is for fuel-oxidizer injection solenoid valve, LED 7 is for ignition relay and LED 8 is for purging solenoid valve. LED blinking test circuit can gives observation on possible operational capability of PDE controller up to around 50 Hz (20 ms) minimum frequency since the human eyes have limited eyesight capability. This minimum frequency is adequate enough to observe the functionality of this PDE controller pinouts for actuators in Table 3.1.

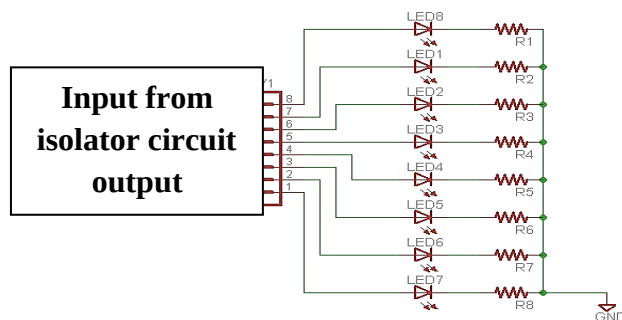


Figure 3.12 : Test circuit schematic for PDE controller.

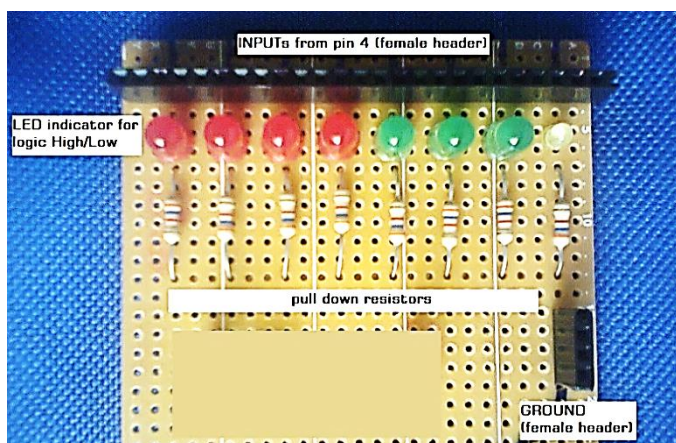


Figure 3.13 : Actual test circuit for PDE controller.

3.6.2 ADC Data Logging Testing and Verification

ADC data logging test and verification consist of 3 parts. They are firmware timing analysis, data logging performance and performance comparison between data capture using *HyperTerminal* versus data logging using USB flash drive. All of these tests were critical to ensure the highest performance throughout the project is achieved and possibilities for future development improvements are discovered.

3.6.2.1 ADC Data Logging Firmware Timing Analysis Method

A test configuration using serial debug interface (UART) was established to collect data for timing analysis through *HyperTerminal* software. Timing analysis was done to determine the critical part of the coding that consumes the most time than others. This is done to understand the behavior of program and to narrow down

the performance optimization options later. By referring to the ADC operation flow chart in Figure 3.11, Figure 3.14 explains the data logging timing in timeframe form.

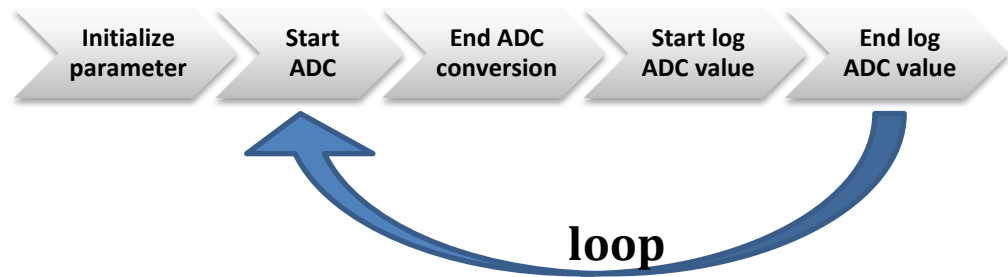


Figure 3.14 : Timing analysis framework.

3.6.2.2 ADC Data Logging Performance Analysis Method

ADC data logging performance analysis were done by logging a dummy ADC input using any voltage source such as VCC of microcontroller into USB flash memory. The data from ADC input was compared with data inside a log file created while logging the ADC values. Based on the timer values and writing counter, a chart of seconds versus number of ADC sample logged per second are plotted. From here we can see the rate of sample logged into USB flash drive per second.

Based on observation in section 3.6.2.1, several different firmware architectures were tested for their performance. The best architecture performance will be implemented in the final prototype.

CHAPTER 4

RESULTS & DISCUSSION

4.1 PDE Controller Analysis

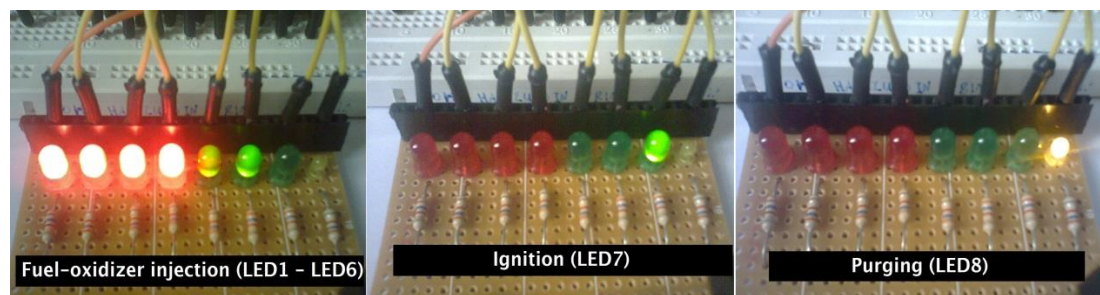


Figure 4.1 : PDE controller operation using test circuit

From observation, all controller pins from isolator circuit output are functioning well as shown in figure 4.1. From figure 4.1, LED1 until LED6 are turned 'ON' while in fuel-oxidizer injection stage. The same goes for LED7 and LED8 for ignition and purging stage.

4.1.1 LED Test Circuit Limitation

The test was done with just only LED array which is far from the actual environment where the relays and solenoid valve is present. These actuators are

using much current rather than LEDs in the test circuit. This would affect the dynamic behaviour of the controller's functionality.

4.2 ADC Data Logging Analysis

ADC data logging analysis is divided into signal attenuator, data logging timing and data logging performance.

4.2.1 Signal Attenuator Linearity Analysis

Signal attenuator for both configurations were tested using potentiometer input voltage (1V, 2V, 3V, 4V, 5V) from a DC power supply. For configuration 1, Table 4.1 shows the result of the test which was plotted in Figure 4.2. On the other hand, the result for configuration 2 attenuator was tabulated in Table 4.2 and plotted in Figure 4.3.

Figure 4.2 shows non-linear result as input voltage is increased using configuration 1 attenuator. This is not suitable for sensor's interfacing with ADC as this condition would lead to false and inaccurate measurement. From Figure 4.3, the plotted result shows linear behaviour of configuration 2 attenuator with respect to the increasing input voltage. This is suitable for sensor's interfacing to provide accurate data for PDE operation.

V_{in}(V)	0.00	1.00	2.00	3.00	4.00	5.00
V_{out} (V)	0.00	0.99	1.98	2.56	2.58	2.59

Table 4.1 : Attenuator configuration 1 result

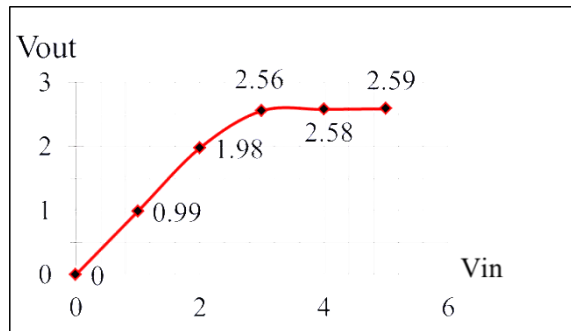


Figure 4.2 : Attenuator configuration 1 plot

Vin (V)	0.00	1.00	2.00	3.00	4.00	5.00
Vout (V)	0.00	0.67	1.21	1.72	2.27	2.93

Table 4.2 : Attenuator configuration 2 Result

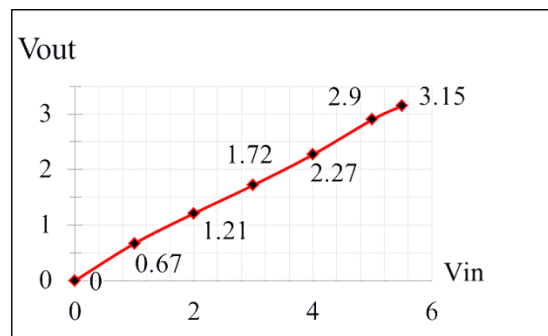


Figure 4.3 : Attenuator configuration 2 Plot

Despite attenuation mechanism had successfully made the pressure transducer's signal output compatible with the ADC input maximum voltage, calibration for the voltage value with respect to the pressure (psi) rating need to be made. Without the attenuator, the maximum voltage input from the transducer is 5.43V at 5000 psi which is 1.086 mV for each psi. A maximum input voltage of 5.5V is assumed rather than 5.43V to design simplify design for signal attenuator. Thus, a simple calculation is needed to calibrate the reading. Table 4.3 tabulates the comparison between before and after attenuation.

$$(5.43\text{V}) / (5.5\text{V}) = 0.9872727 = X / 3\text{V}$$

$$X = 2.961818\text{V} \equiv 5000 \text{ psi (maximum reading)}$$

$$X/(5000 \text{ psi}) = \mathbf{0.6 \text{ mV/psi}}$$

ADC Reading Calibration		
Properties	Before Attenuation	After Attenuation
Smallest voltage resolution detected by LPC1769's ADC	0.8mV	
Max input voltage (at 5000 psi)	5.43 V	3.0 V
Voltage resolution at 1 psi	1.086 mV	0.6 mV

Table 4.3 : Calibration comparison between before and after attenuation

4.2.2 Data Logging Timing Analysis

Marking	Data Logging Position
(a)	Start ADC.
(b)	End ADC conversion.
(c)	Start log ADC value and timer stamp.
(d)	End log ADC value and timer stamp.

Table 4.4 : Data logging position markings.

Micro second time stamp interval (us)		
(a) → (b)	(b) → (c)	(c) → (d)
1136	1130	24,781
1135	1135	26,768
1135	1137	28,759
1136	1139	24,757

Table 4.5 : Timing analysis at random data line.

From the data logging using dummy ADC input, several timing data was extracted at random location within the data lines which is shown in Table 4.5. Around 2,400 data was sampled and time stamped at major coding part (eg: start ADC, end ADC, start log, end log) as in Figure 3.14.

From the data, the average time for data logging is around 26,266 us per data sample or 38 data samples per second. In other words, it consumed 92% of the program loop cycle. Thus, the most time consuming part of the program is from position (c) until position (d) which is from ADC data logging started until end of ADC data logging. Full Speed class USB can actually achieve a maximum of 12Mbit/s data exchange rate. The slow speed happens here is not the problem on USB peripheral in LPC1769 or USB flash memory, but rather a problem in the program itself. According to Chan, a way to optimize the performance of data exchange is by writing multiple sectors per block data in FAT File System. Figure 4.4 shows visualization of writing multiple sectors per block compared to single sector per block. This understanding on performance leads the project to focus on optimizing data writing speed.

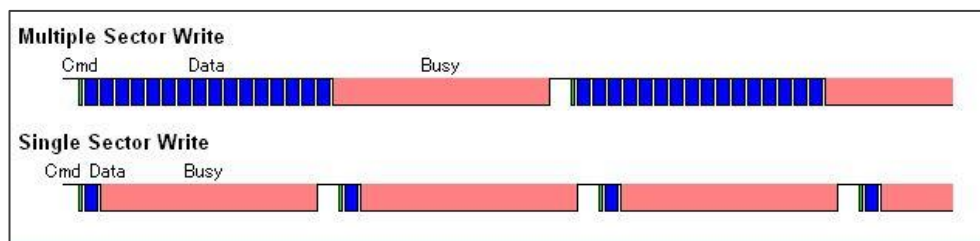


Figure 4.4 : Comparison of multiple sector and single sector per block writing [16].

4.2.2 Data Logging Performance Analysis

When ADC data logging firmware was running, a log file contained parameters is created. The parameters can be programmed to be logged into USB flash drive. An example of logged data sample from test 1 is shown in figure 4.5. From the figure, the first value is the ADC sampling values followed by data writing counter and micro second timer stamp. Figure 4.6 until Figure 3.10 show the results of five different data logging architectures.

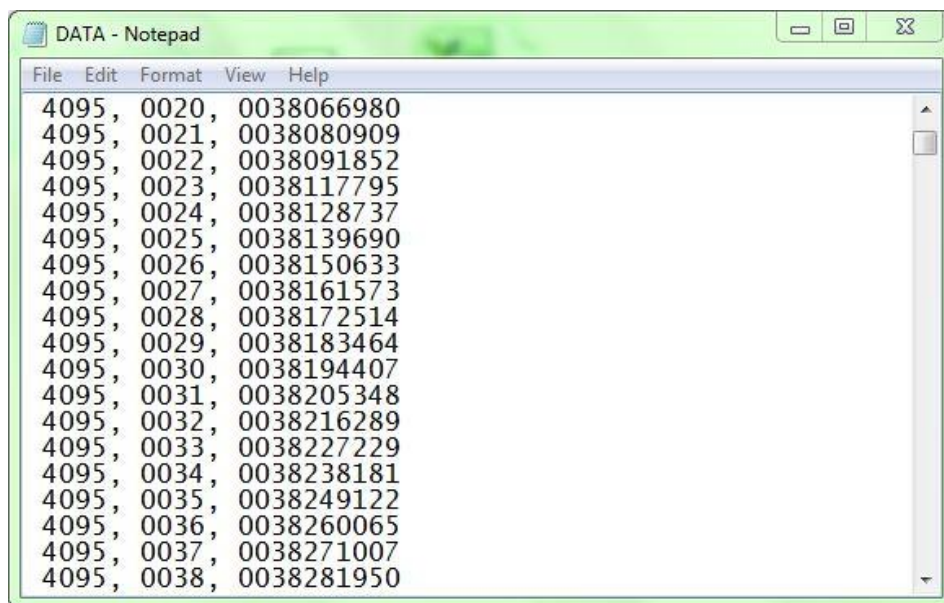


Figure 4.5 : Data logging file example.

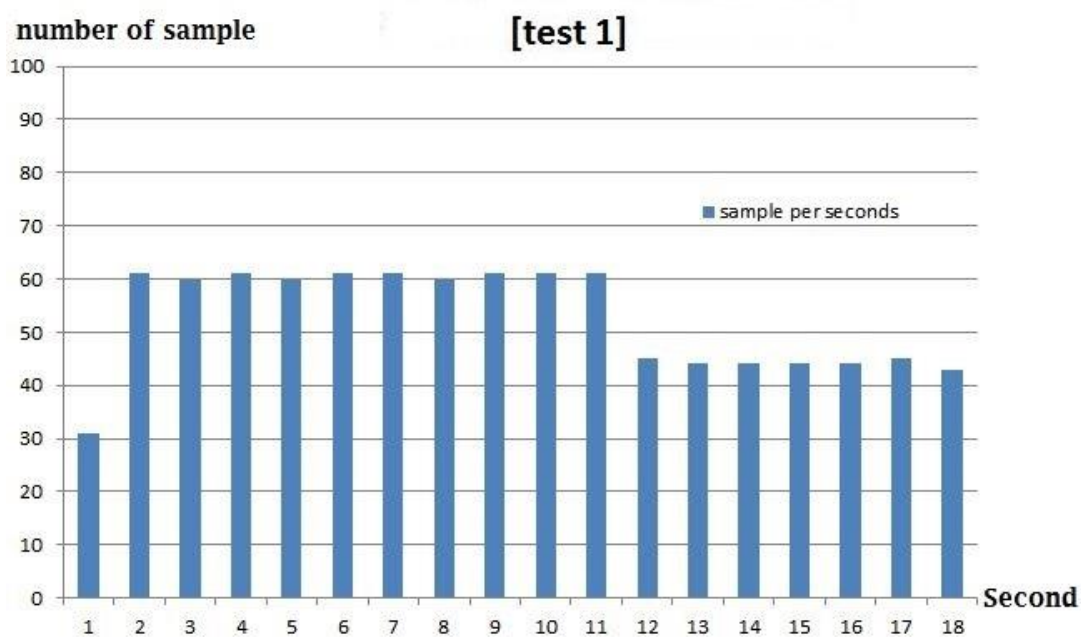


Figure 4.6 : Test 1 ADC data logging result

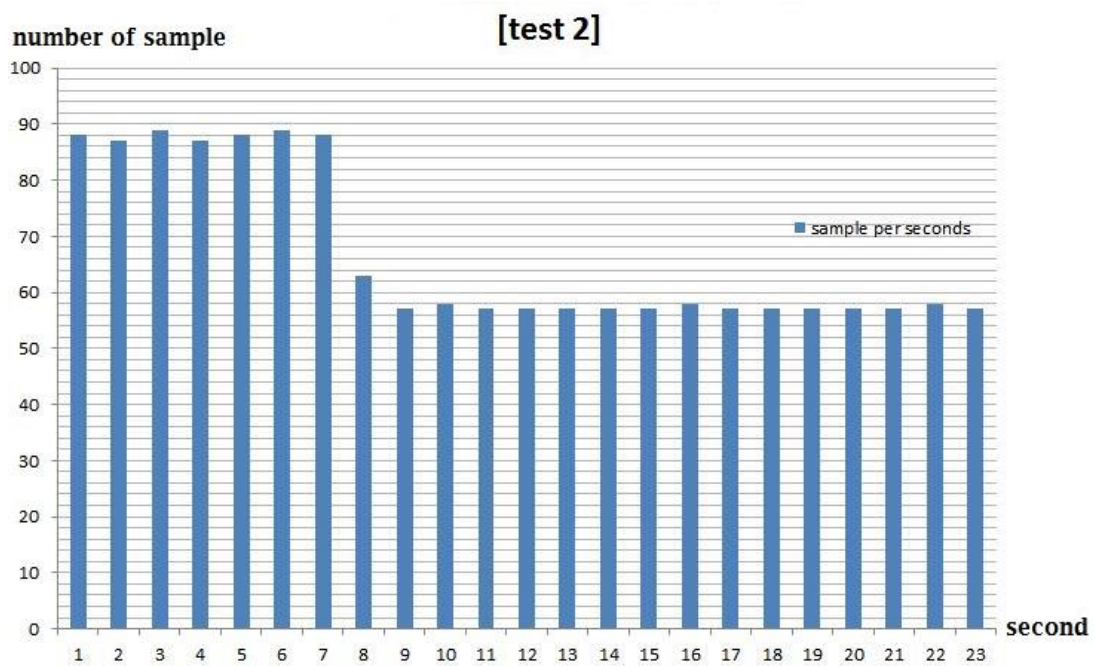


Figure 4.7 : Test 2 ADC data logging result

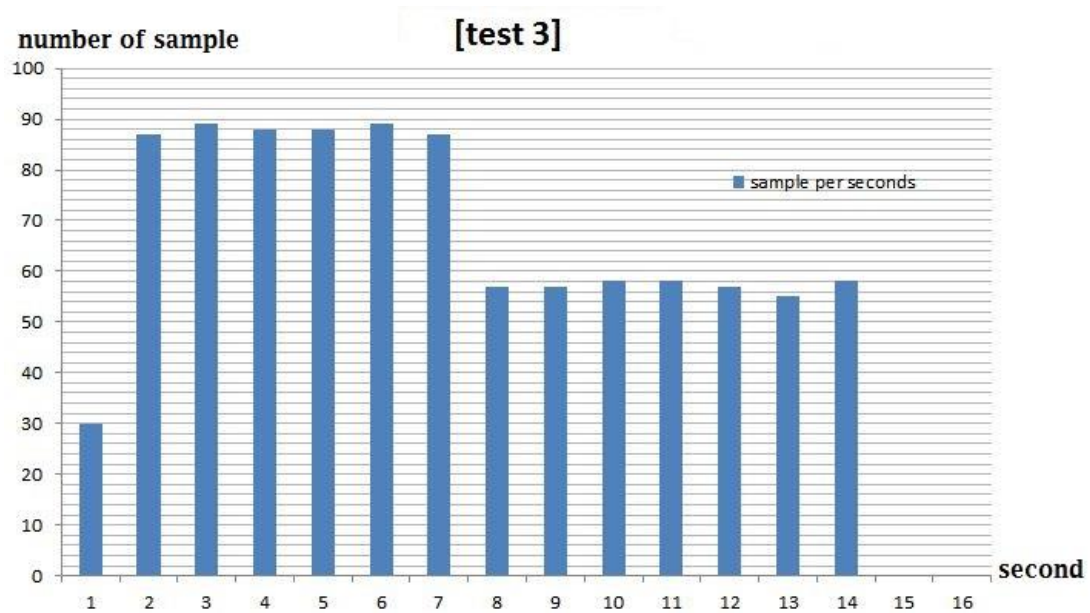


Figure 4.8 : Test 3 ADC data logging result

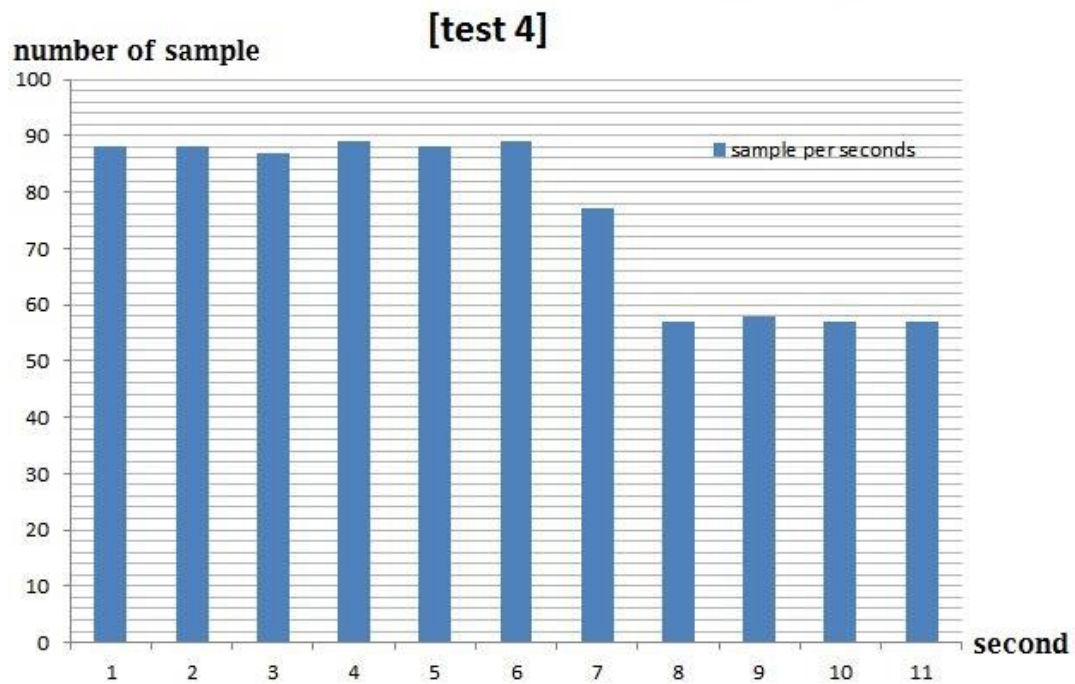


Figure 4.9 : Test 4 ADC data logging result

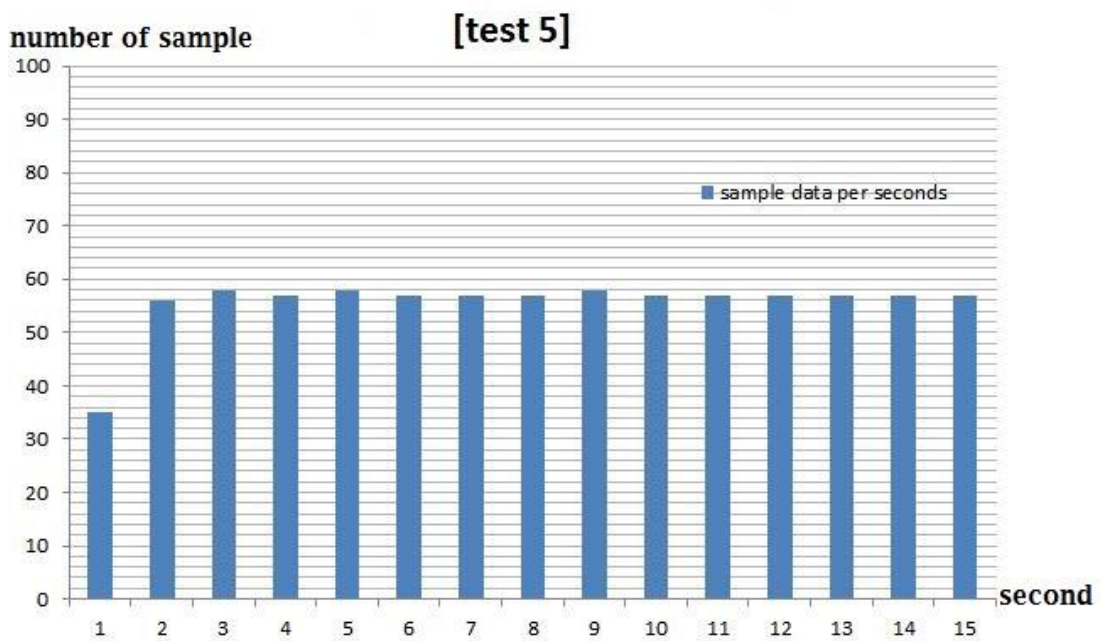


Figure 4.10 : Test 5 ADC data logging result

Figure 4.6 until Figure 4.10 shows number of sample per second versus seconds at which the data was logged into USB flash memory. From the results, we can see a pattern at which the performance of all tests will experiencing rapid performance depreciation after 7th to 11th seconds of data logging. Test 5 shows a

consistent performance but it has low performance average which is around 60 samples per second. Test 1 has the worst average performance because the maximum sample per second it can log is around 60 while the minimum is around 45. Thus, code architecture for test 1 and 5 is filtered out for their low performance among five architectures.

That leaves only test 2, 3 and 4 to analyse with. The best performance for all three of them are at the 7th to 8th seconds which is around 88 sample per second. Test 4 performance starts to fall at the 7th seconds while test 3 and 2 at the 8th seconds. Thus, we can say that test 3 and 2 has the best performance since it can consistently maintain high performance until the first 8 seconds. Despite that, test 3 shows low performance for the first second which means it has high performance data logging for only 6 seconds unlike 7 seconds for test 2. Overall, test 2 has the best ADC data logging performance among all five tests using USB flash memory. Table 4.6 shows the summary of data logging performance analysis.

Marking	Characteristic	Test				
		1	2	3	4	5
(a)	Average number of sample per second (first 8 second)	56	84	76	82	55
(b)	Average of highest rate	60	88	88	88	57
(c)	Number of seconds of highest rate [refer (b)]	10	7	6	6	>20
(d)	Average of lowest rate	45	57	57	57	57
(e)	Number of seconds of highest rate [refer (c)]	>20				
(f)	Average of highest & lowest rate [(b + d) / 2]	52.5	72.5	72.5	72.5	57

Table 4.6 : Summary of data logging performance analysis

4.2.3 UART-based Data Capture Performance Analysis & Comparison

Debug feature in most microcontroller using a terminal emulation software can be applied for getting ADC sampling data using microcontroller. It provides the opportunity to compare the data acquiring performance using terminal emulation software and portable data logging in this project. The format of file created as well as the data arrangement in it is the same as in Figure 4.5.

Figure 4.11 shows the result from ADC data captured through UART serial communication and text capture feature in *HyperTerminal* software. From the result, the data was captured at around 470 samples per second. This performance level was tremendously higher when compared with ADC data logging using USB flash memory storage. Unfortunately, the data need to be saved manually using personal computer or laptop.

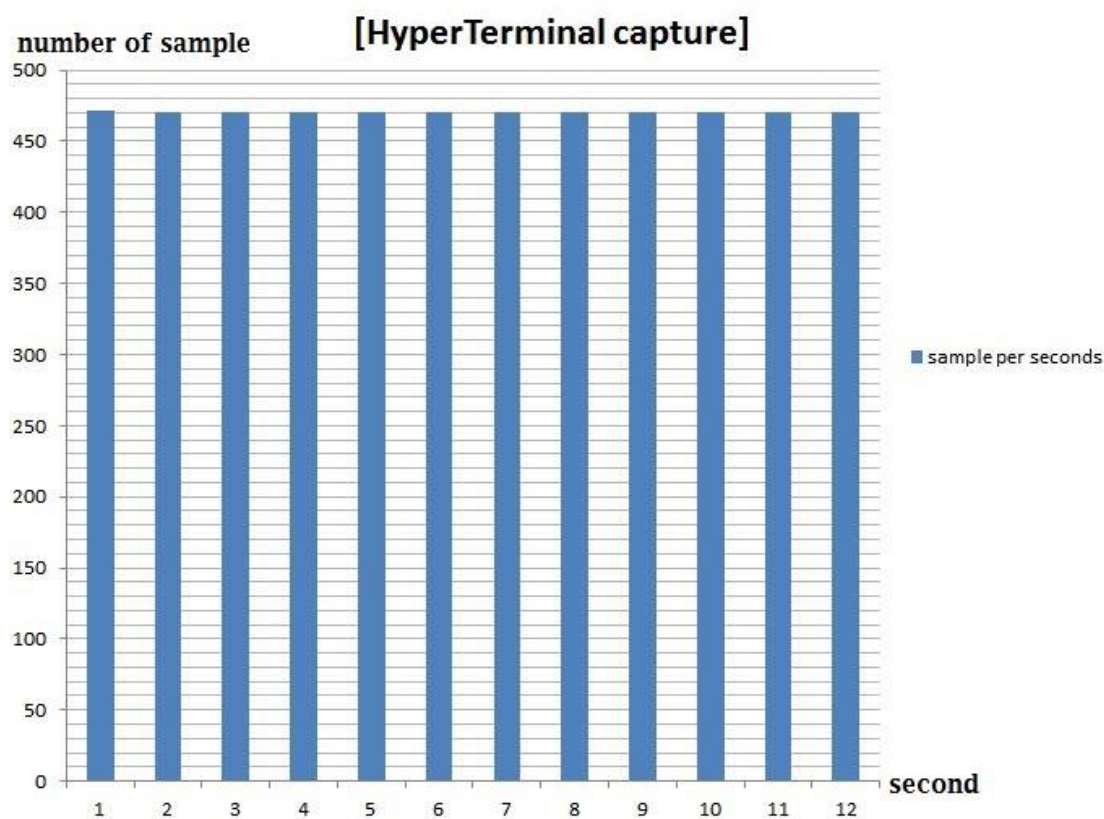


Figure 4.11 : ADC data capture using *HyperTerminal*

4.3 Final Project Prototype

A prototype for this project was made as an example product for this project. This prototype may also be tested in FKM's HiREF laboratory using the Pulse Detonation Engine developed there. Soldering works and wiring were made on donut board based on schematics in previous chapters. Figure 4.12 shows the prototype.

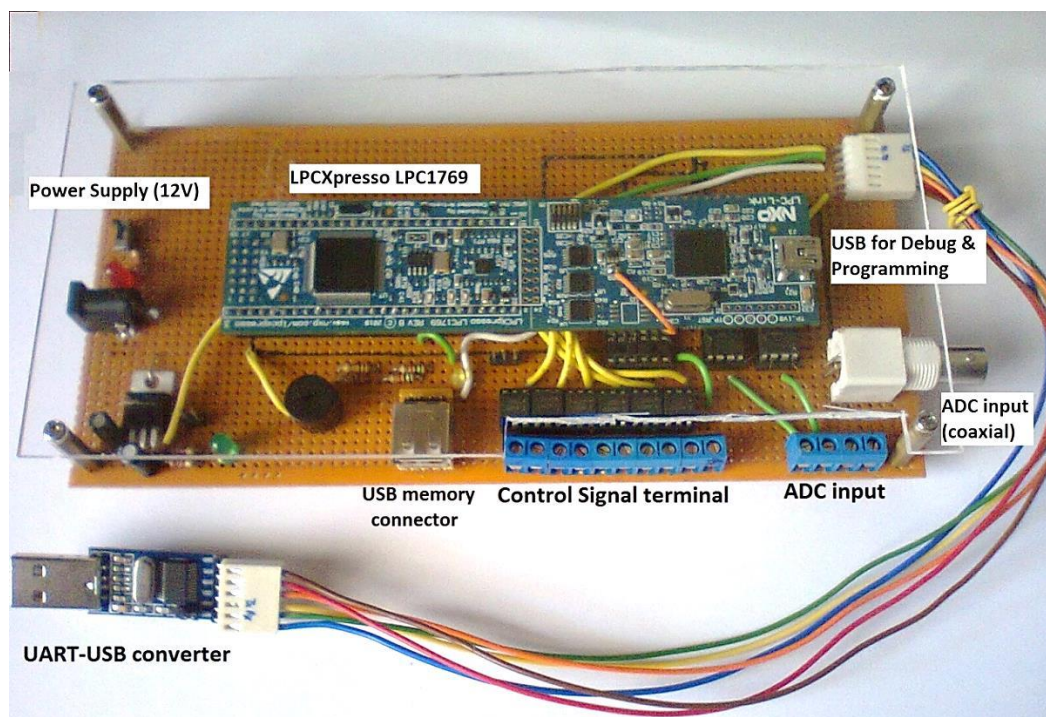


Figure 4.12 : Final project prototype

4.3.1 Limitation of Final Prototype

This prototype has a limitation regarding to this project where the PDE controller and ADC data logging module for this project are separated into different system. In other words, this prototype can only be used as ADC data logger or PDE controller one at a time based on which firmware is programmed into microcontroller. In order to implement the whole system which consist of controller and data logging module, two microcontrollers is needed or two set of final prototype is required.

CHAPTER 5

PROJECT MANAGEMENT

The management aspects for this project were divided into three major parts which are market survey, time management and financial management.

5.1 Market Survey

The market survey was done by using NABC (Need, Approach, Benefit, Competition) approach to justify the marketability factor of this project. The main method used for gathering information on market survey is through articles, journal paper and observation.

5.1.1 Need

The engine is still in research in many institutions. The research-based application need a cost efficient controller and data logger that can be re-configure many times as the research process is progressing. Moreover, the controller needs to be compatible with specific PDE design configuration. Furthermore, there is still no standard controller for pulse detonation engine either for research or commercial purpose available in the market.

5.1.2 Approach

The project is using microcontroller (electronic) based approach that is integrated with electrical-based actuators and analog sensors to control the operation and to read the operation parameters respectively. Most microcontrollers available in the market were provided with open source compiler using C language for hardware programming. C language provides the opportunity to use standard libraries for standard peripheral devices such as USB (Universal Serial Bus), RS232 port and SD memory card.

5.1.3 Benefit per Cost

Detonation Engine operation can be controlled via mechanical or electrical/electronic approach. Currently, the available mechanical approach is using high speed F1 technology valves (up to 20,000 rpm) which are expensive and too complex for PDE application [2]. Thus, electrical/electronic based approach is much cheaper and easy to re-configure using electronic-based controller.

5.1.4 Competition

Currently, there is no significant competition yet since the application is mostly still in research phase in most institutions.

5.2 Time Management

The time management for this project was done using a working timeline for FYP1 and FYP2 separately. The timeline was sectioned into weeks for FYP1 and FYP2 as provided in Electrical Engineering Final Year Project Logbook. Some phases of the project may have different allocated weeks than others since the expected time required to complete the phases are significantly different.

Work		Week															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Understand the project																
2	Studying LPC1769 MCU and LPCXpresso v6.02																
3	Configuring ADC module																
4	Configuring RS232 interface																
5	Testing ADC and UART on PDE's sensors																
6	FYP1 report writing																

Figure 5.1 : Working timeline for FYP1

Work (FYP2)		Week																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	Establish/configure ADC - UART communication																	
2	Verifying ADC and UART communication																	
3	Identifying/configure controller mechanism																	
4	Verifying controller																	
5	Testing the controller with sensor on-board PDE																	
6	Top-level testing																	
7	FYP2 report writing																	
8	Report draft submission with journal paper																	
9	Hardbound report submission																	

Figure 5.2 : Working timeline for FYP2

5.3 Sourcing Management

Many elements such as end-user for Pulse Detonation Engine, component, hardware integration, testing equipment, and firmware development need to be source out to execute this project. Sourcing management is divided into several segments which are testing equipment, electronic components/tools, firmware development, end-user consultation and teammate/colleague.

5.3.1 Testing Equipment

The testing equipments used in this project are digital oscilloscope, digital multimeter and PC/laptop. Table 5.1 shows tabulation of equipment with their respective source and usage.

Equipment	Source	Usage
Digital oscilloscope	Embedded System Laboratory, Faculty of Electrical Engineering	- Observe the behaviour of signal from pressure transducer for PDE. - Determine linearity of input signal versus output signal of a custom-designed signal attenuator for pressure transducer and microcontroller's ADC interfacing.
Digital multimeter	- Embedded System Laboratory, Faculty of Electrical Engineering - Self-own	- Continuity testing for soldered components on donut board or microcontroller development board. - Troubleshoot errors in circuit connection.
PC/Laptop	Self-own	- Writing reports, outsource information. - Using IDE for developing, debugging and programming microcontroller's firmware. - Simple GUI (Graphical User Interface) for serial communication.

Table 5.1 : Equipment sourcing tabulation.

5.3.2 Electronic Components/Tools

Most electronic components were sourced (bought) from *Cytron Technologies* in Taman Universiti, Skudai, Johor. Due to unavailability of rare or outdated electronic components in the market, they are sourced (bought) at

Hitechtron in Johor Bahru and via online ordering at *Element14* (formerly Farnell). The components bought at *Cytron Technologies* and *Hitechtron* were immediately received on the same day of buying while online ordering via *Element14* takes around 3 days for the ordered components to arrive. Thus, a Bill of Material (BOM) has been developed for this project as in Table 5.2 below.

Bill of Material (BOM)			
Cytron Technologies Sdn Bhd			
No	Component Name	Model/ brand/ model number	Quantity
1	LED 5mm	-Red and green	8
2	Op-amp	-LM741	6
3	Optocoupler	-4N25 -Black package	8
4	IC socket (8 pin)	-	6
5	IC socket (6 pin)	-	10
6	SOIC/SSOP to PID adapter	-Cytron	3
7	Jumper wires (male-male)	-Cytron	2 packet
8	DB9 male (RS232)	-	2
9	RS232 cable (male-male)	-	1
10	UART to RS232 converter	-Cytron	1
11	UART-USB converter	-Cytron	1
12	Potentiometer (small)	-10k ohm -N6	4
13	Potentiometer (big)	-10k ohm	1
14	Wire	-Single core	3 m
15	Wire	-Multi core	3 m
16	Donut Board (small)	-6x15cm	3
17	Terminal block	-2 way	7
18	Straight Pin Header (male)	-1x40 ways	2
19	Straight Female Header	-1x40 ways	2
20	Resistors	-Standard -Various value	~40
21	Capacitors	-0.1uF -10uF	~10
22	Buzzer	-5V -PCB mount	1
23	USB connector	-Type A	1
24	Mini slide switch	-	1

Element14			
25	3.3V RS232 transceiver	-SOIC type. -Analog Device	1
Hitechtron Sdn Bhd			
26	Coaxial pin	-Male	1
27	Connector (socket ,cable)	-2510	1 set

Table 5.2 : Bill of material (BOM) for this project

Some components/tools don't need to be bought since it's already owned or provided by supervisor which are tabulated in Table 5.3 below.

Components/Tools	Source
LPCXpresso LPC1769 Development Board	- Embedded System Laboratory, Faculty of Electrical Engineering.
Solder wire, solder wick, solder gun, solder paste	- Embedded System Laboratory, Faculty of Electrical Engineering. - Self-own.

Table 5.3 : Components/Tools readily available

5.3.3 Firmware Development

Development of embedded system using microcontroller platform leads to firmware or code outsourcing. Each microcontroller may have their own unique architecture or peripheral libraries. Usually, the manufacturer will provide access to the generic microcontroller peripheral libraries through internet download.

For LPC1769 microcontroller, there is at least one major source for embedded system development which is *LPCware.com*. This website is being supported by NXP Semiconductors and further developed to cater the needs for outsourcing and feedback. It provides download access for peripheral's example libraries, microcontroller's standard USB peripheral driver, user manuals and application notes. This site also provides discussion forum where developers around

the world can ask any enquiry regarding embedded system design using NXP microcontrollers. Furthermore, it provides registration service for LPCXpresso IDE users.

5.3.4 End-User Consultation

This project was intended for Pulse Detonation Engine end-user at FKM's HiREF laboratory. The requirements, specifications and problems may need to be consulted in-person with the end-user.

On Tuesday, 26th November 2013 a site-visit to FKM's HiREF laboratory was made. The intention of site-visit was to study and understand the operation mechanism of HiREF's Pulse Detonation Engine. From the site-visit, problems and limitations was discovered in testing the controller and ADC data logging where the person-in-charge was not always available for assistance. Furthermore, there are also problems in the electrical circuitry for PDE's actuator circuit where the earlier controller circuitry was poorly made and the capacitor for ignition circuit was always breaks down. Thus, it is hard to debug and troubleshoot possible circuitry problems. Some suggestion on testing alternatives was made to test the pressure transducer without having to operate the PDE such as building a customizable pressure device which can simulate variable pressure conditions in detonation tube.

5.3.5 Teammate/Colleague

This Final Year Project was under Mr Zulfakar supervision. There are other students under him which uses the same microcontroller platform (LPCXpresso LPC1769). Hands-on introductory class for LPCXpresso IDE and LPC1769 microcontroller application was made together with the students under supervision of Mr Zulfakar ex-student. They are Chairul Ramadan Bin J Effendy and Siti Haslina Bt. Furthermore, we discuss among ourselves about problems and firmware building regarding LCP1769.

5.4 Financial Management

Since this is a hardware-based electronic project, a lot of components and equipment may need to be bought at local stores or be ordered online. This process will contribute to the cost of the hardware to be produced.

Basically, the bought project components can be referred from Top Level Graphical Representation of the Project in Figure 3 and Bill of Material (BOM) in Table 5.2. Thus, the cost of the components was tabulated in the Table 5.4.

No	Component Name	Model/ brand / model number	Quantity	Price/ quantity (RM)	Total Price (RM)
Cytron Technologies Sdn Bhd					
1	LED 5mm	-Red and green	10	0.10	1.00
2	Op-amp	-LM741	8	1.00	8.00
3	Optocoupler	-4N25 -Black package	10	2.00	20.00
4	IC socket (8 pin)	-	8	0.20	1.60
5	IC socket (6 pin)	-	10	0.20	2.00
6	SOIC/SSOP to PID adapter	-Cytron	3	5.00	15.00
7	Jumper wires (male- male)	-Cytron	2 packet	4.50	13.50
8	DB9 male (RS232)	-	2	1.80	3.60
9	RS232 cable (male-male)	-	1	28.00	28.00
10	UART to RS232 converter	-Cytron	1	20.00	20.00
11	UART to USB converter	-Cytron	1	19.00	19.00
12	Potentiometer (small)	-10k ohm -N6	4	0.50	2.00
13	Potentiometer (big)	-10k ohm	1	1.20	1.20
14	Wire	-Single core	3 m	0.40	1.20
15	Wire	-Multi core	3 m	0.90	2.70
16	Donut Board (small)	-6x15cm	3	1.20	2.40
17	Terminal block	-2 way	10	0.70	7.00
18	Straight Pin Header (male)	-1x40 ways	4	0.60	2.40
19	Straight Female Header	-1x40 ways	4	1.20	4.80

20	Resistors	-Standard -Various value	~40	0.05	2.00
21	Capacitors	-0.1uF -10uF	~10	~ 0.20	2.00
22	Buzzer	-5V -PCB mount	1	3.00	3.00
23	USB connector	-Type A	1	1.80	1.80
24	Mini slide switch	-	1	0.80	0.80
25	Voltage regulator	-5V, 1A		1.00	2.00
Element14 (formerly Farnell)					
26	3.3V RS232 transceiver	-SOIC type. -Analog Device	1	9.72	29.16
Hitechtron Sdn Bhd					
27	Coaxial pin	-Male	1	5.90	5.90
28	Connector (sockets ,cables)	-2510	1 set	4.80	4.80
Total = RM 206.86					

Table 5.4 : Hardware development costing

CHAPTER 6

CONCLUSION & SUGGESTION

6.1 Conclusion

Based on the objective of this project, two conclusions were made. First, a controller for Pulse Detonation Engine has been designed successfully and the timing parameter of the detonation cycle can be changed until micro second interval accuracy using interrupt timer peripheral of LPC1769 for all stages in PDE operation cycle. However, the controller is still in testing prototype level and need to be integrated with the actual Pulse Detonation Engine in FKM's HiREF laboratory for further testing and verification.

Second, the data logging module can successfully log ADC input signal using USB flash memory storage. The highest performance for the data logger is around 88 samples per second.

6.2 Suggestion for Future Improvement

First, the system can utilize Interrupt peripherals on-board LPC1769 to simplify both PDE controller and data logging module so that the system will only use a single microcontroller for this system. This will greatly reduce the cost for system implementation.

Second, the core platform for this project (LPCXpresso 1769) can be changed into other industrial standard ARM-based platform. There are a lot of manufacturers for microcontroller or microcomputer that implemented ARM architecture into their design. LPC1769 has limited development opportunity for non-expert embedded system developer such as undergraduate students since there is very limited resources can be found for embedded system development. Moreover, This project focuses on ADC data logging based on USB flash memory and general purpose I/O application. There are many other options for development platform such as PIC series from Microchip Technology Inc, Freescale Semiconductors and Atmel can be further explored to suit effectively with this project's requirement, especially for high speed data logging purpose. Moreover, Linux-based microcomputer such as Raspberry Pie and Beaglebone may be implemented for its excellent GUI (Graphical User Interface) and networking capability.

Second, the performance can be increased by further deeply exploring FAT File System for external flash memories such as USB flash memory, SD memory card and external hard disc drive. The capability to use this memory media in the highest performance will lead to greater possibility in embedded system design that requires high speed and high volume data storage.

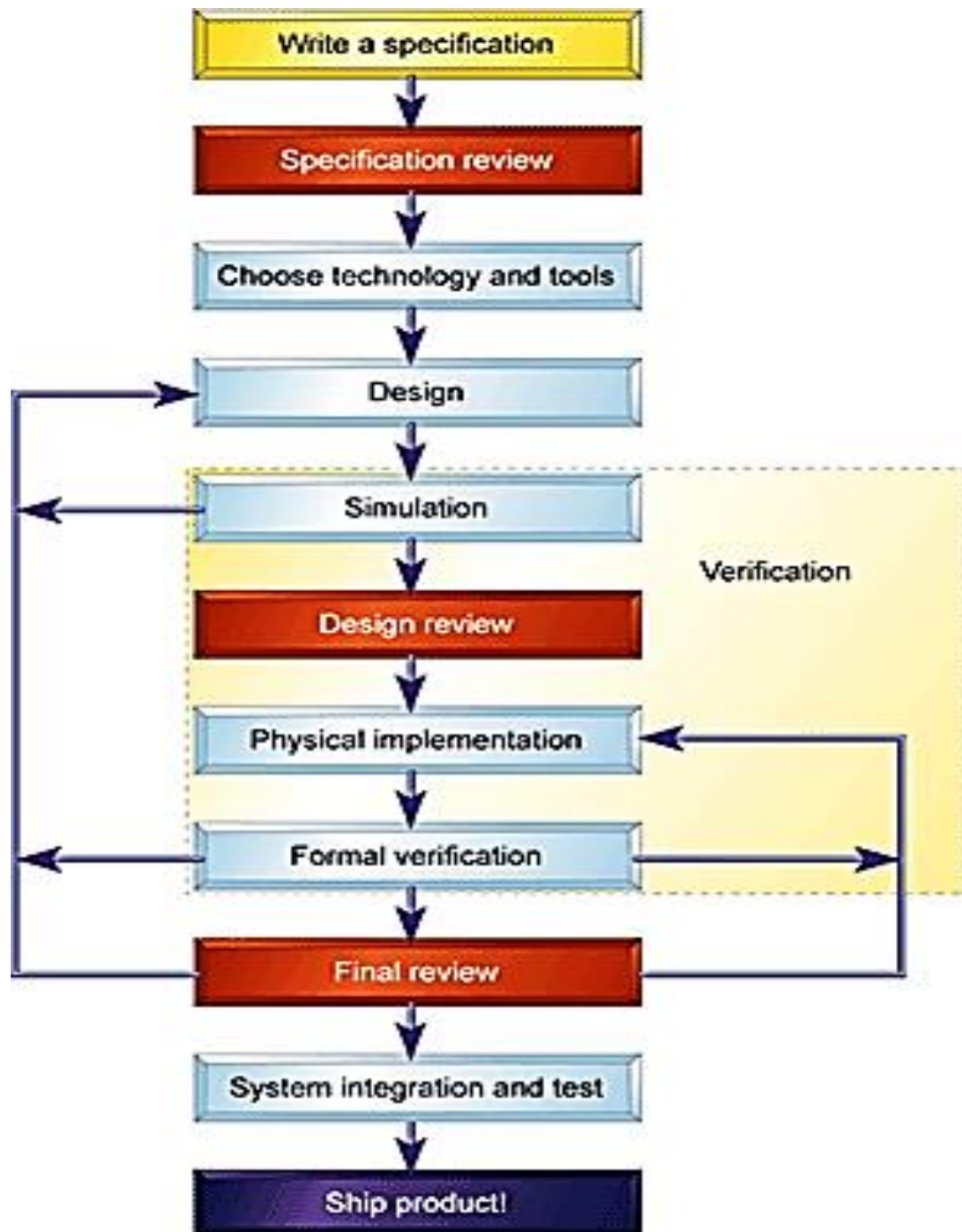
Third, to increase user friendly and marketability property for this project, a GUI to operate the prototype or final product need to be developed. *Visual Basic* or *Qt* development software can be used for this particular purpose.

Fourth and last, management matters with FKM's HiREF laboratory may also be improved to synchronize all testing, requirements and specification matters. This is very important since this project is related to other institution within UTM premise. The success of project or research for both parties is detrimental to UTM's image and capability expectation by other people.

References

1. Khalid M. S., Ahmed F., H. K., Mohsin S., Mazlan A.W. “**Transient Characteristics of C₃H₈/O₂ Turbulent Mixing in a Hypersonic Pulse Detonation Engine**,” Proceedings of the 9th WSEAS International Conference on APPLICATIONS of COMPUTER ENGINEERING.
2. Philip K. Panicker, Donald R. W., Frank K. Lu. “**Operational Issues Affecting the Practical Implementation of Pulsed Detonation Engines**,” *AIAA Paper* 2006-7959, 14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference.
3. P. F. Seitz and R. F. Searle. “**Space Shuttle Main Engine Control System**,” 730927, National Aerospace Engineering and Manufacturing Meeting Los Angeles, California., October 16-18, 1973.
4. Shmuel Eidelman, Xiaolong Yang. “**Analysis of the Pulse Detonation Engine Efficiency**”. Science Applications International Corporation, 1710 Goodridge Drive, McLean VA 22102.
5. Thomas Bussing, George Pappas. “**Pulse Detonation Engine Theory And Concepts**”. ASI (Adroit Systems, Inc.), Bellevue, Washington 98004.
6. NXP Semiconductors. “**UM10360, LPC17xx user manual, Rev. 2**”. 19 August 2010.
7. Microchip Technology Inc. “**3V Tips ‘n Tricks**”. 2006.
8. <http://www.embedded.com/electronics-blogs/beginner-s-corner/4024888/The-universal-design-methodology>
9. Fairchild Semiconductor Corporation. “**General Purpose 6-Pin Phototransistor Optocouplers, 4N25**”. 6 June 2002.
10. NXP Semiconductors. “**LPC1769/68/67/66/65/64/63, Product data sheet, Rev. 9.3**”. 8 January 2014.
11. NXP Semiconductors. “**AN10974, LPC176x/175x 12-bit ADC design guidelines, Rev. 1**”. 1 September 2010.
12. Ahmad Faiz B. Mad Zin. “**Development of Pulse Detonation Engine**,”. 2011. Universiti Teknologi Malaysia, Master of Engineering (Mechanical) Thesis.
13. <http://www.cytron.com.my/>
14. <http://arm.com/products/processors/cortex-m/index.php>
15. <http://www.nxp.com/>
16. http://elm-chan.org/fsw/ff/00index_e.html
17. http://www.embeddedartists.com/products/lpcxpresso/lpc1769_xpr.php
18. <http://www.lpcware.com/>

APPENDIX 1

Appendix 1 : Universal Design Methodology (UDM) from www.embedded.com